A Supervised Learning Approach for Imbalanced Text Classification of Biomedical Literature Triage

Hayda M. S. Almeida

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

April 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Hayda M. S. Almeida**

Entitled: **A Supervised Learning Approach for**

Imbalanced Text Classification of Biomedical Literature Triage

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair

Dr. C. Constantinides

_____ Examiner

Dr. Gregory Butler

_____ Examiner

Dr. Brigitte Jaumard

_____ Co-supervisor

Dr. Leila Kosseim

_____ Co-supervisor

Dr. Marie-Jean Meurs

Approved by _____

Chair of Department or Graduate Program Director

_____ 2015 _____

Amir Asif, Ph.D. Dean

Faculty of Engineering and Computer Science

ABSTRACT

A Supervised Learning Approach for
Imbalanced Text Classification of Biomedical Literature Triage

Hayda M. S. Almeida

This thesis presents the development of a machine learning system, called *mycoSORT*, for supporting the first step of the biological literature manual curation process, called triage. The manual triage of documents is very demanding, as researchers usually face the time-consuming and error-prone task of screening a large amount of data to identify relevant information. After querying scientific databases for keywords related to a specific subject, researchers generally find a long list of retrieved results, that has to be carefully analyzed to identify only a few documents that show a potential of being relevant to the topic. Such an analysis represents a severe bottleneck in the knowledge discovery and decision-making processes in scientific research. Hence, biocurators could greatly benefit from an automatic support when performing the triage task. In order to support the triage of scientific documents, we have used a corpus of document instances manually labeled by biocurators as "selected" or "rejected", with regards to their potential to indicate relevant information about fungal enzymes. This document collection is characterized by being large, since many results are retrieved and analysed to finally identify potential candidate documents; and also highly imbalanced, concerning the distribution of instances per relevance: the great majority of documents are labeled as rejected, while only a very small portion are labeled as selected. Using this dataset, we studied the design of a classification model to identify the most discriminative features to automate the triage of scientific literature and to tackle the imbalance between the two classes of documents. To identify the most suitable model, we performed a study of 324 classification models, which demonstrated the results of using 9 different data undersampling factors, 4 sets of features, and the evaluation of 2 feature selection methods as well as 3 machine learning algorithms. Our results demonstrated that the use of an undersampling technique is effective to handle imbalanced datasets and also help manage large document collections. We also found that the combination of undersampling and feature selection using Odds Ratio can improve the performance of our classification model. Finally, our results demonstrated that the best fitting model to support the triage of scientific documents is composed by domain relevant features, filtered by Odds Ratio scores, the use of dataset undersampling and the Logistic Model Trees algorithm.

Acknowledgments

Firstly, I would like to express my deepest gratitude to my supervisors Dr. Leila Kosseim and Dr. Marie-Jean Meurs. Their guidance, willingness to help and constant enthusiasm to share their knowledge contributed inestimably to my progress during my academic experience. It was an honor to have had the opportunity to work with these two wonderful supervisors, whom I profoundly admire for the outstanding researchers and professors they are. I am extremely greateful for their endless patience, consideration, kindness and understanding. I feel truly privileged to have received their support and I could not have imagined being guided by better mentors.

I would like to thank Dr. Adrian Tsang for the support provided to develop this research. Thank you for embracing and believing in my research project.

I am also greateful to the examining committee, Professors Gregory Butler and Brigitte Jaumard, as well as the chair Professor Constantinos Constantinides, for their helpful feedback and insightful comments.

Many thanks to my fellow labmates at the Centre for Structural and Functional Genomics (CSFG), as well as at the Computational Linguistics at Concordia (ClaC), for their support, help, interesting discussions and valuable contributions during these two years of research.

I am also thankful for

Finally, I want to dedicate this thesis to my parents, who offered me unconditional encouragement and conviction in my potential. Words cannot to express my gratitude and love. Your support was my greatest incentive to achieve this goal.

Contents

List of Figures **vii**

List of Tables **viii**

1 Introduction **1**

1.1 Problem Context and Research Motivation 1

1.2 Problem Statement 2

1.2.1 Imbalanced Data 3

1.2.2 Feature Selection 4

1.3 Our Contributions 5

1.4 Thesis Outline 6

2 Literature Review **7**

2.1 Classification of Biomedical Documents 7

2.2 Imbalanced Learning Techniques 9

2.2.1 Cost-Sensitive Learning 10

2.2.2 Data Sampling Methods 11

2.3 Feature Selection Techniques 14

2.3.1 Inverse Document Frequency (IDF) 16

2.3.2 Odds Ratio (OR) 17

2.4 Classifiers and Imbalanced Data Approaches 17

3 Experiment Methodology **19**

3.1 Dataset 20

3.2 Training and Test Corpora 21

3.3 Corpus Sampling 22

3.4 Document Representation 26

3.4.1 Feature Extraction and Types 26

3.4.2 Feature Selection Strategy . . . 28
3.4.3 Feature Vector . . . 29
3.5 Classification Algorithms . . . 29
3.5.1 Naïve Bayes . . . 30
3.5.2 Logistic Model Tree . . . 30
3.5.3 Support Vector Machine . . . 32

4 System Evaluation **34**
4.1 Evaluation Metrics . . . 34
4.2 Experimental Setup . . . 37
4.2.1 Set of Features . . . 37
4.2.2 Classifiers . . . 38
4.2.3 Undersampling . . . 38
4.3 Experimental Results . . . 39
4.4 Discussion . . . 42
4.4.1 Most Discriminative Features . . . 43
4.4.2 Imbalanced Learning Strategy . . . 46
4.4.3 Best Feature Selection Method . . . 48

5 Conclusion and Future Work **56**
5.1 Main Findings . . . 56
5.1.1 Scientific Contributions . . . 56
5.1.2 Other Contributions . . . 58
5.2 Availability and Reproducibility . . . 59
5.3 Future Work . . . 60

References **62**

A Dataset Queries **71**

B *mycoSORT* Experimental Results **73**

List of Figures

1 Pipeline of the *mycoSORT* system 3
2 Balance of positive and negative instances in *mycoSet* training sets using undersampling 23
3 Algorithm used to perform the *mycoSet* test set sampling 24
4 Algorithm used to perform the *mycoSet* training sets undersampling 25
5 Overview of a Logistic Model Tree in a binary classification 31
6 Separation of data performed by a SVM classifier given a hyperplane H 33
7 Computation of Precision (P) and Recall (R) using TP, FP and FN instances . . . 36
8 *mycoSORT* F-2 scores for the baseline and the best model for the approach using only USFs (TM1) 40
9 *mycoSORT* F-2 scores for the baseline and the best model for the approach using USFs and Odds Ratio (TM3) 41
10 Summary of *mycoSORT* F-measure for the positive class. Best classifiers and set of features for each USF 43
11 Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF 44
12 Summary of *mycoSORT* F-measure scores for the positive class. Best classifiers and set of features for each USF using Inverse Document Frequency 50
13 Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF using Inverse Document Frequency 50
14 F-2 scores of baseline and best model for the approach using USFs and Inverse Document Frequency 52
15 Summary of *mycoSORT* F-measure for the positive class. Best classifiers and set of features for each USF using Odds Ratio 54
16 Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF using Odds Ratio 54

List of Tables

1 Cost matrix of a binary classification 10
2 Statistics on the *mycoSet* corpus 21
3 Percentage of instances across progressive undersampling of the *mycoSet* training data 23
4 The 22 bioentities and spans annotated in *mycoSet* by the mycoMINE text mining system 27
5 *mycoSet* sample feature vector representing feature occurrences for one document . . 29
6 Confusion matrix of a binary classification task 34
7 Summary of all of *mycoSORT* tables of experimental results 41
8 Summary of all of *mycoSORT* charts of experimental result 42
9 *mycoSORT* results obtained for the baseline approach 42
10 Comparison of the positive class scores for models using sets of features S3 and S4 for the lowest and highest USFs 44
11 Comparison of the positive class scores for models using sets of features S1 and S3 for the highest USFs 45
12 Comparison of the positive class scores for models using sets of features S1 and S2 for the lowest and highest USFs 46
13 Comparison of the positive class scores for models using 0% USF and 40% USF with the LMT algorithm 47
14 Number of features in the models before and after applying feature selection methods 48
15 Comparison of the positive class scores for models using Inverse Document Frequency and models with no feature selection 49
16 Comparison of positive class scores of models using Odds Ratio and models with no feature selection 53
17 Comparison between best results obtained from the different model design approaches 55
18 *mycoSORT* results using USFs for set S1 74
19 *mycoSORT* results using USFs for set S2 75
20 *mycoSORT* results using USFs for set S3 76

21 *mycoSORT* results using USFs for set S4 77
22 *mycoSORT* results using USFs for set S1 + Odds Ratio filtering 78
23 *mycoSORT* results using USFs for set S2 + Odds Ratio filtering 79
24 *mycoSORT* results using USFs for set S3 + Odds Ratio filtering 80
25 *mycoSORT* results using USFs for set S4 + Odds Ratio filtering 81
26 *mycoSORT* results using USFs for set S1 + Inverse Document Frequency filtering . 82
27 *mycoSORT* results using USFs for set S2 + Inverse Document Frequency filtering . 83
28 *mycoSORT* results using USFs for set S3 + Inverse Document Frequency filtering . 84
29 *mycoSORT* results using USFs for set S4 + Inverse Document Frequency filtering . 85

Chapter 1

Introduction

1.1 Problem Context and Research Motivation

Biomedical publications are an essential information source for the knowledge discovery and decision-making process of scientific researchers. Scientific documents are usually maintained in massive databases that nowadays grow exponentially, following the pace of publications of scientific findings. Over the past few years, researchers and users have noted a significant expansion of such literature databases [Hunter and Cohen, 2006]. As of January 2015, the public on-line database PubMed ([National Center for Biotechnology Information, 2005a]) held over 24 million documents, and a simple keyword search on PubMed could retrieve more than hundreds of thousands of documents. For example, in January 2015[1], if a user was looking for fungal related information from this database and queried the string `fung*`, almost 250,000 documents would have been returned. Another example is a search to query information about the HIV virus, which using the string `HIV`, could provide the user with almost 280,000 results. It is precisely because of their massive content that research databases are vital resources for scientists. In addition, they allow storing information in a consistent way, facilitating easy retrieval and enabling both complex searches and computation on data.

However, to keep up with the continuous updating of the knowledge discovery process, researchers are frequently retrieving and analyzing new data from these databases in a time-consuming and generally not exhaustive manner. Several tools and approaches have been developed to assist scientists in the manual analysis of documents currently in the literature. Analyzing and handling the vast biomedical data available is an important challenge that has been addressed by various studies (e.g., [Kasprzyk et al., 2004] [Wang et al., 2014] [Morris and White, 2013] [Smith et al., 2012]), as well as the application of this data to identify relevant information in biomedical research

[1] At the time of writing this thesis.

(e.g., [Mudunuri et al., 2013] [Quan et al., 2014]).

The automatic classification of biomedical texts has been utilized in various contexts to support researchers in identifying important information in science papers, and release the burden of manually reviewing large amounts of data. In particular, supervised learning approaches are valuable to support biomedical literature screening, since they can help scientists to evaluate a greater number of documents in a shorter period of time. Such approaches can also reduce the possibility of missing relevant documents, since an automatic (system-based) screening might be less error-prone than if the same task is performed manually [Wallace et al., 2010].

The workflow of bioliterature curation is divided in five main steps [Hirschman et al., 2012]: finding relevant documents, identifying relevant entities in these documents, annotating and encoding relevant events, associating experimental evidences and finally inserting curated information in a database. The first step, called "triage" of documents, represents a severe bottleneck in the process [Howe et al., 2008] [Lu and Hirschman, 2012] [Hirschman et al., 2012]. The triage is a demanding task for scientific researchers, as it requires to manually identify documents with curatable content among an extensive list of papers that were retrieved by a query search. The triage of biomedical documents is an example of biomedical text classification that could greatly benefit from an automatic approach, as previously demonstrated by [Wallace et al., 2010] and [Lu and Hirschman, 2012]. In this thesis, we propose a system to support the triage process, by automatically classifying these documents, after having learned from a correctly labeled set of documents. The pipeline of our proposed system, called *mycoSORT*, is shown in Figure 1. First, a collection of manually labeled documents is used to extract discriminative features. Next, a classification model is designed with the acquired information, and then used to predict the output of new document instances, that have not yet been seen by the *mycoSORT* system.

1.2 Problem Statement

Text classification to support manual biomedical literature triage can pose great challenges to the development of an automatic approach. This thesis addresses two main problems that are characteristic of this task. The first problem is the imbalanced data, caused by the discrepancy between the class distribution in the dataset. The other challenge discussed is feature selection, a method that is necessary to this task in order to reduce the feature space size, remove noisy attributes and reduce the task computational cost, as in the time taken to fit a classification model. These issues will be briefly introduced in the following sections.

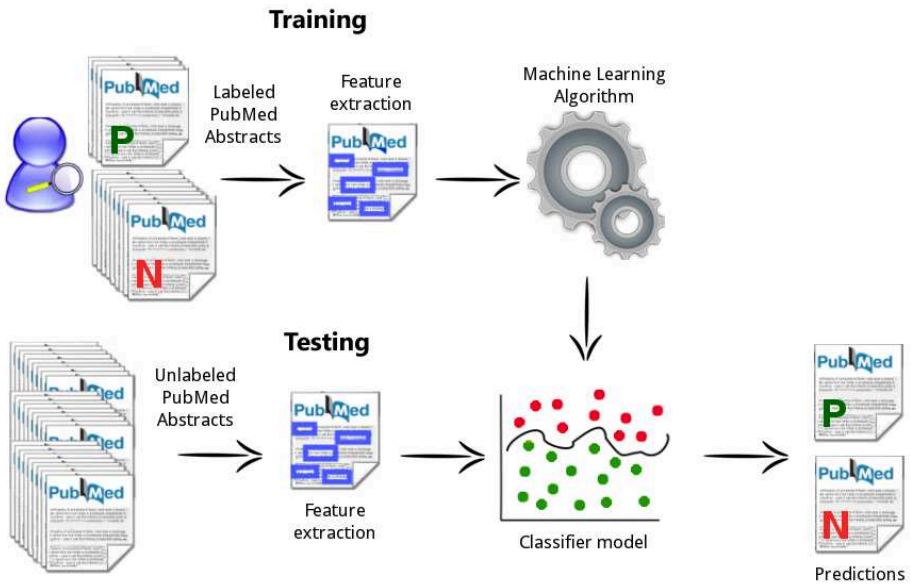

Figure 1: Pipeline of the *mycoSORT* system

1.2.1 Imbalanced Data

A dataset is said to be imbalanced if it presents a skewed distribution of instances among its classes. When a dataset is imbalanced, the difference in the number of instances belonging to each class is so severe that it affects the effectiveness of the prediction output of a machine learning process [He and Garcia, 2009]. A document collection that is representative of the biomedical literature triage task will be highly imbalanced, since the number of documents identified as curatable represents only a small fraction of the documents that are analyzed. In the context of this research, curators are looking for scientific articles related to characterized lignocellulose-active proteins of fungal origin that will populate the mycoCLAP database [Murphy et al., 2011]. The presence of relevant documents is limited to an average representation of only 10% of the total set of retrieved documents.

The imbalance between the curatable and non-curatable documents in the collection represents a hardship for classification algorithms. When dealing with a non-balanced dataset, the mathematical properties of various classification algorithms lead the model construction to a bias. Several classifiers will tend to favor the majority class and overlook the minority class instances, since they will try to maximize the overall accuracy.

As [Raskutti and Kowalczyk, 2004] noted, the problem of machine learning from imbalanced data is common in many real world applications beyond biomedical text classification. The imbalance issue is also studied, for example, in classification problems related to fraud detection (e.g., [Fawcett and Provost, 1997][Bolton and Hand, 2002]), medical diagnosis (e.g., [Antonie et al., 2001][Cohen

et al., 2006]) and speech recognition (e.g., [Liu et al., 2006]).

Imbalanced learning can interfere directly on the classifier performance. Because the majority class is more represented in the data than the minority class, it tends to have more influence under uncertainty cases since the class distribution can affect the learning criteria. In addition, according to [Weiss and Provost, 2001], a classifier presents a lower error rate when classifying an instance belonging to the majority class, since it will have learned more information from the examples of the majority class, compared to the information learned in fewer examples from the minority class.

Since classifiers usually tend to maximize accuracy, the misclassification errors are equally considered. This implies that a majority instance when misclassified as a minority one will have the same error cost that a minority instance misclassified as a majority one. Because in tasks with imbalanced dataset the minority class is so little represented, even if a classifier assigns the majority class label to all minority instances, the overall accuracy would still be fairly acceptable. Therefore, a high accuracy measure in an imbalanced context does not demonstrate that the classification model is capable of clearly identifying the minority class instances which are generally the most relevant to the task.

1.2.2 Feature Selection

As numerous documents are available and retrieved after a search, the triage task dataset is extensive. Using a naïve approach to extract features from a large dataset, will most likely produce a large feature space. This characteristic can raise at least two difficulties. First, because a great number of features can be extracted from these documents to be used in the model design, the classifier learning process will then have a high computation cost, making it longer to learn the classification function. Second, using a large and sparse feature space might induce the model to overfit the data, and thus present a weak performance when classifying new input information.

Feature selection is defined as the task of eliminating less relevant or redundant features from a given set. Methods of feature selection have been applied mostly with the purpose of handling the high dimensionality of classification models. Various approaches have been proposed to select features in an informed manner without influencing the model learning [Liu and Motoda, 2007].

Extensive classification models are commonly generated in tasks that handle large datasets. As [Liu et al., 2010] [Saeys et al., 2007] noted, the use of feature selection approaches can help improve the model classification performance. First, system performance can be improved since having a smaller number of features will result in a more compact model, demanding less computational resources in the learning phase. Second, classification performance can also be improved with feature selection, since some techniques aim to reduce the model dimension without losing useful information.

This can be achieved, for instance, by discarding noisy attributes that could interfere in the decision-making process. Eliminating non-discriminative features and maintaining a more concise feature space can also reduce the probability of model over-fitting and contribute to a better classification performance.

Studies, such as [Saeys et al., 2007] [Peng et al., 2010] [Haury et al., 2011], introduced feature selection methods commonly applied in biomedical data. Already dealing with a large dataset and an extensive feature space, the learning process of the triage task can become even more expensive if other methods are plugged in during the model building phase. Filtering methods used for feature selection have the advantage of being executed as a pre-processing step to the learning phase. This means that the process of selecting features can be independently incorporated in the pipeline, without influencing other existing data processing methods, such as data sampling. Also, because it is executed before the learning phase, feature filtering does not add computational cost in the model construction, since by then, the selected subset of features to be used is already defined, and no extra time is required to re-process features during the learning phase.

As explained in the beginning of this section, imbalanced data and feature selection are two common problems characteristic to the task of biomedical literature triage and can affect directly the performance of a machine learning system. In order to design a classification model that has enough discriminative power to recognize the potential relevant instances, we studied and presented state-of-the-art research in Chapter 2, while in Chapter 3 we describe the approach we have adopted to tackle the triage task and design a model capable of mostly outputting correct predictions for the important documents.

1.3 Our Contributions

This thesis offers a contribution to the study of biomedical text classification. More specifically, it introduces a supervised learning approach to provide automatic support to curators of bio-literature, cutting human effort down to a considerable extent. The main contribution of this work is the full implementation of the *mycoSORT* system, a software that is capable of classifying documents according to their relevance to a given topic.

In order to develop *mycoSORT*, we thoroughly evaluated and compared 324 classification models, designed based on different data sampling and feature selection techniques, and by means of a detailed analysis of our experimental results, we have demonstrated the combination of methods yielding the best performances. Thus, another contribution of this thesis is the design and implementation of the most suitable classification models to handle the task of biomedical literature

triage.

1.4 Thesis Outline

In this chapter, we presented the context of our work and our motivation, as well as a brief introduction to the problems that are being addressed by our research. We also have summarized the scientific contributions provided by this thesis. The next sections of this thesis are structured as follows: **Chapter 2** presents an overview of the previous work conducted to address imbalanced learning approaches, feature selection techniques and algorithms applied in imbalance learning tasks. We start by discussing the two most popular techniques used in tasks handling imbalanced datasets, that are cost-sensitive and data sampling. We then introduce two well-known feature selection metrics, Inverse Document Frequency and Odds Ratio. Next, we provide a review of the machine learning algorithms commonly applied in imbalanced learning scenarios. **Chapter 3** describes in detail the methodology adopted to design and conduct our experiments. First, an explanation of the corpus creation and composition is given, including the preparation of training and testing datasets and the undersampling approach adopted. Secondly, we provide a brief discussion of the framework implemented to represent document instances with regards to the features extracted and selected by our feature selection strategy. Finally, we briefly describe the properties of the three classification algorithms used in our experiments. In **Chapter 4** we discuss the details of the system evaluation. We start by introducing the metrics used to assess our results. Then, we describe in detail the experimental settings adopted for our designed classification models, that were derived from the combination of the approaches described in Chapter 3. Finally we present and discuss the results obtained by our classification models. **Chapter 5** provides a conclusion of our work, by analyzing our main findings regarding the data undersampling and feature selection techniques when applied to the task of scientific literature triage, and we conclude by presenting possible future work.

Chapter 2

Literature Review

In this chapter, we will provide a review of the most relevant previous work conducted in the three following areas: imbalanced learning techniques, feature selection methods and classification algorithms for imbalanced data. We start with Section 2.1, that gives an overview of automatic approaches to classify biomedical documents. In Section 2.2, we present the state-of-the-art imbalanced learning techniques. Section 2.3 introduces feature selection techniques, including a review of methods specifically applied on tasks using biomedical datasets. Finally, in Section 2.4 we present an overview of the machine learning algorithms most commonly applied in imbalanced learning tasks.

2.1 Classification of Biomedical Documents

The development of an automatic approach to classify biomedical literature has been previously investigated and evaluated in several studies. The most important effort in the area is the BioCreative[1] initiative [Hirschman et al., 2005], a challenge dedicated to assess text mining and information extraction approaches for biological data. In particular, the third and fourth BioCreative challenges [Matis-Mitchell et al., 2013], along with the 2012 edition [2] [Arighi et al., 2013], addressed several biomedical text classification tasks.

[Kim and Wilbur, 2011] and [Ambert and Cohen, 2012] described literature triage approaches implemented for solving the BioCreative III tasks, organized in 2010, in the track of identifying protein-protein interactions [Arighi et al., 2011] [Krallinger et al., 2011]. [Kim and Wilbur, 2011] addressed the dataset imbalanced issue related to scientific document classification. In a task where almost 83% of the dataset is composed by non-relevant instances, this study described that state-of-the-art approaches for imbalance learning did not improve their performance results. To achieve one

[1] http://www.biocreative.org/

[2] The BioCreative challenges are sequentially identified by Roman numerals, while the BioCreative workshop organized in 2012 was not assigned a sequence number.

of the top performances in the task, the authors used syntactic patterns and word features combined to design a classification model, and demonstrated that the use of domain features can positively influence system performance. To deal with the imbalanced class distribution, [Ambert and Cohen, 2012] also described an interesting approach that compared the similarity between two documents using a k-Nearest Neighbors (k-NN) classifier based on the Information Gain value of their common features. This approach led to the best performance among all participating systems at BioCreative III.

A web-based system to support the biomedical triage was introduced by [Hsu and Kao, 2013], during the BioCreative-2012 workshop, in the track of document ranking for curation. This approach was based on the computation of co-occurrence of features and co-occurrence of networks formed by named entity pairs, composed by gene-disease, gene-chemical and chemical-disease, which were extracted from the different documents. The higher the co-occurrence of pairs, the higher the likelihood of a paper to be curatable.

For the BioCreative IV challenge workshop, organized in 2013, [Campos et al., 2014] and [Kwon et al., 2014] presented systems capable of performing biomedical literature triage in the track of interactive curation [Arighi et al., 2014], that requested participants to present web-based systems to perform any of the tasks in the biocuration workflow. Although the systems participating in this track were able to perform literature triage, the task was more focused on the software usability aspects, and their ability to meet a broader spectrum of manual biocuration steps, than to provide new models to tackle the specific issues related to the biomedical literature triage. [Campos et al., 2014] introduced a web-based system to support different steps of the manual biocuration process including the triage of papers, which takes into account the relevance of protein-protein interactions extracted in each document instance. [Kwon et al., 2014] presented a system that implemented a supervised learning framework to rank documents according to the relevance of protein-protein interaction features. The document relevance is computed based on the previously top-ranked approach presented by [Kim and Wilbur, 2011]. Another system to perform biomedical text classification was introduced by [Romero et al., 2014]. The system is able to execute classification of relevant and non-relevant documents, as well as perform attribute selection and dataset sampling.

However, all tools and BioCreative IV systems described above presented an approach that is fairly generic. These systems are more focused on providing users with a more off-the-shelf solution, based on a large set of generic supporting tools for biomedical literature classification tasks. Differently from these generic approaches, the focus of our project is to address the specific issues of the triage task of the manual biocuration process, and design a problem-oriented classification model. The evaluations conducted in the designed models will analyze their capability of handling the triage of documents containing lignocellulose-active proteins of fungal origin. Our approach

focuses in addressing specifically the issues of the imbalanced class distribution and the extensive feature space size, that are discussed in Sections 2.2 and 2.3.

2.2 Imbalanced Learning Techniques

A dataset is considered imbalanced if, in a document collection, a class is so little represented that the definition of a classification function can be compromised during the learning process [He and Garcia, 2009]. Generally, document collections that support biomedical research are expected to present a highly imbalanced characteristic, since discovering document instances that are revelant to a given topic (positive class) is much less common than discovering instances not relevant to this topic (negative). The use of an imbalanced dataset to learn a classification function can affect directly the classifier performance in identifying instances belonging to the relevant, and therefore less represented, class. This behavior occurs because the mathematical properties of classification algorithms usually aim to maximize the overall accuracy, which leads to a rather undesirable behavior, when the classifier shows an accuracy of around 100% for negative (and more common) instances, at the cost of performing poorly in the positive (less common) instances [Chawla et al., 2002].

Therefore, in an imbalanced scenario, special attention must be given when choosing the evaluation metrics to assess the performance of a classifier built based on imbalanced data. Using a single measure such as accuracy to evaluate the model performance does not clearly indicate if the most relevant instances are being correctly classified. On one hand, if only accuracy is considered as a performance measure for imbalanced classification, the classifier can easily achieve very high scores, simply by considering all instances to belong to the majority class. On the other hand, the minority instances, that probably represent the task target, will be missed. In this work, we will make use of five evaluation metrics that are more suitable to evaluate the performance of tasks that handle imbalanced data. These metrics are presented in Chapter 4.

The state-of-the-art research, summarized by [He and Ma, 2013], describes five main techniques to tackle the imbalance dataset problem: sampling, cost-sensitive, kernel-based learning, active learning and one-class learning methods. The two most popular techniques are the cost-sensitive technique and the sampling technique. The sampling technique is described as the most common technique applied at the data level, while the cost-sensitive technique is the most commonly applied at the algorithm level [He and Ma, 2013]. Sampling is a procedure that allows to balance the number of instances pertaining to different classes in a document collection. Depending on the sampling method applied, document instances can either be removed from or added to the collection. Differently, the cost-sensitive technique tries to assign different weights to classification errors made at the minority class compared to errors made at the majority class, helping to fit a more appropriate function.

Several comparative studies (e.g., [Weiss et al., 2007] [McCarthy et al., 2005] [Chen et al., 2004]) have evaluated the performance of cost-sensitive against sampling techniques. Both techniques have demonstrated performance improvement when compared to a naïve or baseline approach. However, no results were conclusive to show the superiority of one technique to deal with imbalanced class distributions. Neverthless, [Weiss et al., 2007] pointed out that the class distribution found in an imbalanced dataset is an important condition to be considered when designing a classifier model, since it can affect the efficiency of a particular technique. Both cost-sensitive and sampling techniques are further described in Sections 2.2.1 and 2.2.2, and a brief comparison of their characteristics is provided.

2.2.1 Cost-Sensitive Learning

The cost sensitive learning technique, described in [Elkan, 2001][Maloof, 2003], applies a different weighting criterion for the classifier error cost computation. This strategy can help improve the classification performance, since in an imbalanced scenario, some errors are considered more costly than others. If an instance belonging to the most common class is classified as a minority instance, it may not constitute an important mistake. However, the performance can decrease significantly if instances belonging to the minority class are classified as if they belonged to the most common class. Since the rare class is usually the one of interest for biomedical text classification tasks, this indicates that the most valuable information is being overlooked by the classifier.

Cost-sensitive classifiers aim to minimize classification errors of the rare class by biasing the classifier towards making mistakes in the common class instead. The cost-sensitive strategy tries to compensate the class imbalance by defining a cost matrix for classification errors. This cost matrix assigns different weights to incorrect predictions, according to the output class and the actual class. For binary classification tasks, a cost matrix can be implemented as shown in Table 1.

	Actual Negative	**Actual Positive**
Classified Negative	Cost(0,0) = $c_{0,0}$	Cost(0,1) = $c_{0,1}$
Classified Positive	Cost(1,0) = $c_{1,0}$	Cost(1,1) = $c_{1,1}$

Table 1: Cost matrix of a binary classification

In the cost matrix, the sum of the columns (Actual Negative + Actual Positive) represents the actual number of instances a dataset contains in each class; whereas the sum of the rows (Classified Negative + Classified Positive) indicate the number of instances predicted for each class. Each classification algorithm will process the assigned costs of a matrix in a different manner, but the weights defined in a cost matrix are incorporated in the algorithm during the learning phase, and

not as a pre-processing step. According to [Elkan, 2001] [Weiss et al., 2007], not all classification algorithms can handle the cost-sensitive technique, since the properties of some algorithms do not allow threshold adjustments when computing an instance class prediction, and are only capable of reproducing the predictions according to the underlying characteristics of the data. This can make the cost-sensitive approach restrictive to deal with the imbalanced issue. [Elkan, 2001] explains that a given threshold can be incorporated in the decision-making process of a classifier, so that when the algorithm is computing the predictive value of a certain instance, the threshold can introduce a cost for incorrectly classifying positive instances. For example, using the cost matrix shown in Table 1, the prediction cost p^* for a cost-sensitive classifier is:

$$p^* = \frac{c_{1,0}}{c_{1,0} + c_{0,1}} \tag{1}$$

However, not all algorithms can calculate a precise posterior probability estimation as the prediction cost p^*, since they are only capable of computing the exact classification output.

2.2.2 Data Sampling Methods

Sampling was described in [He and Ma, 2013] as the most popular technique used to deal with imbalanced datasets. The technique consists of selecting a specific subset of the available population to be taken into account for the training phase. The selection of a data subset in the sampling technique is done either randomly or in an informed manner.

A common benefit brought by the use of data sampling is the low computational cost, because the data processing is executed in a pre-learning phase and therefore does not add extra processing time in the model building. In fact, several studies (e.g., [Maloof, 2003][Borrajo et al., 2011]) have compared the use of sampling methods and the application of other techniques to handle imbalance datasets. Although sampling has not being shown to provide a better performance with regards to other techniques, compared to them it was demonstrated to be more flexible. Some techniques to handle imbalance data, such as the cost-sensitive technique, present limitations that prevent them from being applied in combination with certain classifiers. Although sampling is a less restrictive method in this sense, it can bring other limitations, which are discussed below for each sampling approach presented.

Several approaches to perform sampling have been presented in the literature. [Chawla et al., 2002] discussed the two main sampling methods: oversampling and undersampling. The next paragraphs will describe these two techniques along with their advantages and drawbacks.

Oversampling

The oversampling method consists of adding instances belonging to the minority class to the document collection until the number of minority instances reaches a similar number of the majority class instances. As new instances are usually not available, this additional data is artificially generated by randomly replicating existing instances.

An important drawback of this method is the fact that new instances are generated as duplicates of existing instances of the rare class. Oversampling the minority class by synthetically generating new instances is likely to overfit the training set. A classifier that was learned based on this dataset will be too well adjusted to the training data, and as a consequence, will not be capable of generalizing properly the model to predict the output of test instances.

A few studies have analyzed alternatives to implement the oversampling method using an informative approach. In particular, [Japkowicz, 2000] describes the application of the focused resampling method, in which only rare class instances that are found next to the decision boundary are replicated. This approach reduces the amount of synthetic data generated, yet it ensures that only the most relevant data among the rare instances are used. Another approach that addresses the oversampling drawback was introduced by the Synthetic Minority Over-sampling TEchnique (SMOTE), described in [Chawla et al., 2002]. This technique suggests that a synthetic instance should be generated based on the combination of a specific instance and one of its k-nearest neighbors. Many other variations of the SMOTE technique were developed afterwards (e.g., [Bunkhumpornpat et al., 2009] [Bunkhumpornpat et al., 2012] [Ramentol et al., 2012]). Still, in general, oversampling implementations in the literature are based on the generation of artificial data instances in order to reach a more balanced class distribution.

Undersampling

The undersampling method consists of discarding a portion of the most common instances until a certain balance in the class distribution is achieved. Different from oversampling, undersampling is less likely to over-fit to the training set. However, it has a major drawback related to information loss. Important data for structuring the classification model can be missed by the act of discarding instances of the majority class. This information loss interfers in the definition of a more complete decision boundary, since the model will have been deprived from the majority instances not taken into account.

Undersampling can be performed based on random instance selection, or with the use of a more informative approach, as an attempt to reduce the anticipated information loss of the process. Various undersampling approaches were presented in the literature to address the issue of losing data from the majority instances discarded in the process. In particular, [Japkowicz, 2000] introduced a

focused downsize technique, that discards majority instances only found at the farthest points from the decision boundary. [Hoens and Chawla, 2013] described the use of the Neighborhood Cleaning Rule (NCR) [Laurikkala, 2001] to perform undersampling. In this technique, the majority instances selected to be discarded are the ones found surrounded by minority class instances, and therefore considered the most noisy.

On one hand, removing instances from the majority class might represent a loss of information about the dataset. On the other hand, eliminating these instances may result in a simpler model, with fewer features, that if taken into account could introduce noise instead of actually contributing to the classifier discriminative power.

Comparative studies have measured the performance results of oversampling and undersampling methods. Some evaluations demonstrated that the use of undersampling outperformed the oversampling method. In particular, [Drummond and Holte, 2003] and [Luengo et al., 2011] showed that undersampling yields a better performance in tasks using datasets from various domains. [Loyola-González et al., 2013] also conducted an evaluation of both sampling methods using a variety of datasets, in which the results demonstrated that undersampling yields improved performance when the imbalanced ratio is equal or more severe than 1:2.

Previous studies (e.g., [Luengo et al., 2011], [Drummond and Holte, 2003], [Rahman and Davis, 2013], [Estabrooks et al., 2004]) have compared the performance of these two methods in classification tasks. The results did not clearly indicate that one method is more suitable or outperforms the other in general. Sampling methods can be easily implemented in a algorithm-independent manner and require a low computational cost. However, in circumstances where the number of minority instances on the document collection is small, characterizing an absolute rarity, the use of sampling methods might not yield performance improvement or might not even help handling the task at all, as observed by [Estabrooks et al., 2004] and [Weiss, 2013]. Moreover, while the sampling technique is an attempt of reducing the bias on the data towards the minority class, when oversampling or undersampling is applied, a new bias is introduced by these methods [Weiss, 2013]. Adding more instances of the less common class, or removing instances of the more common class causes the learning algorithm now to deal with a disparate class distribution as the one of the underlying problem, which may affect performance during the test phase.

According to [Rahman and Davis, 2013], undersampling is an adequate method to handle biomedical data, typically known to be imbalanced. The effectiveness of undersampling methods has also been assessed in several tasks using biomedical datasets, and hence many studies have applied this method to tackle the imbalanced data issue in biomedical research (e.g., [Tang and Zhang,

2006], [Zhang et al., 2011], [Varassin et al., 2013], [Yu et al., 2013]). Another advantage of undersampling, as noted by [Weiss et al., 2007], is the low computational cost of the learning phase. Since the dataset is reduced by the removal of a bulk of majority instances, the task will require less processing time, and in scenarios where the amount of data exceeds the computational resources available to process it, data undersampling could even make the task feasible.

The great advantage of sampling methods is that they can be conveniently incorporated in a machine learning pipeline and be very useful to handle imbalanced class distribution and large datasets. In addition, further methods were studied to also address the innate computational effort of imbalanced classification tasks. In the next section, feature selection techniques are discussed as a method of improving the overall classification performance and reducing processing costs.

2.3 Feature Selection Techniques

As described in [Guyon and Elisseeff, 2003], feature selection is the process of identifying feature subsets that are effective to a certain task. It can bring many benefits to machine learning systems, as explained by [Liu et al., 2010]. Feature selection will not only allow the feature space dimension to be reduced, requiring less computational resources to process the data, but will also eliminate noisy and irrelevant attributes. This leaves a cleaner set of features available to design the model and reduces the chance of over-fitting, which could negatively affect the classification performance during the test phase.

According to [Saeys et al., 2007] [Liu et al., 2010], feature selection techniques are generally divided in three main categories: filtering methods, embedded methods, and wrapper methods. Filtering methods consider the evaluation of each feature in an independent way, without directly interfering in the classification algorithm. Wrapper methods evaluate the different model hypotheses generated from the use of various possible subsets of features. Embedded methods incorporate within the classifier construction a search for the ideal feature subset for the current task.

Several previous studies (e.g., [Saeys et al., 2007] [Peng et al., 2010] [Liu et al., 2010] [Haury et al., 2011]) have described interesting advantages of the filtering methods, when applied to biological text mining and classification. When compared to wrapper and embedded methods, filtering methods seem to be more advantageous when several techniques are being combined to design a classification model. First, as filtering methods can be executed separately from the classifier learning phase, the entire classification task requires less computational resources to be completed. Second, also because filtering can be performed before the learning phase, no strong dependency is created between the feature selection method and the classifier building, which allows more flexibility to experiment with different model configuration setups.

This last characteristic is beneficial in our current context, since the approach described by our research evaluates the combination of various techniques to cope with both large datasets and feature spaces. Therefore, considering a model-independent feature selection will provide less restrictions in terms of experimental settings and will facilitate the evaluation of the individual performance of each approach used in the classification model design.

In addition, through an evaluation of various feature selection methods, [Haury et al., 2011] has demonstrated that classification results of wrapper and embedded methods did not outperformed results of filtering methods on a task of biological feature selection.

[Wasikowski and Chen, 2010] studied the use of feature selection techniques to tackle the imbalanced dataset issue. The results showed that feature selection can be effective when facing imbalanced class distributions, including in the biological domain. According to the authors, the features might be too sparse to effectively provide a generalization of the distinct classes on the dataset. The authors studied a better feature-document ratio to represent a classification problem when using biological data, and the results yield performance improvement when feature selection techniques were used to reduce the feature space to a size that is similar to the number of instances in the dataset, resulting then in a square matrix (with an equal number of rows and columns). With this assumption, the performance improved independently of the classification algorithm. The datasets used for the experiments of [Wasikowski and Chen, 2010] were rather small, with 60 to 540 instances. Additionally, the authors observed that, although improving the performance with these datasets, using feature selection to handle imbalanced data might not be as effective with larger datasets. For this different context, the authors recommended an approach that evaluates the performance of classification algorithms and data sampling techniques, a similar approach to the one adopted in this thesis.

In the literature, many feature selection metrics have been applied to text classification tasks. The most popular ones include:

- Information Gain (IG), as the criterion applied in Decision Trees algorithms [Quinlan, 1986];
- Chi-Square test (CHI), that evaluates the dependence between a given term and a class;
- Term Frequency (TF), which represents the raw frequency of a term in the entire dataset;
- Document Frequency (DF), that accounts for the number of documents containing a term, calculated for each single feature across all instances in the dataset;
- Inverse Document Frequency (IDF), that computes the number of documents containing a term, weighted by all documents in a collection;
- Odds Ratio (OR), which computes the likelihood of encountering a term given a specific class.

Feature selection metrics have been compared in various studies (e.g., [Yang and Pedersen, 1997] [Forman, 2003] [Almeida et al., 2011] [Basu and Murthy, 2012]). [Yang and Pedersen, 1997] described the Document Frequency (DF) as a simple, reliable and effective measure to evaluate feature relevance. [Forman, 2003] analyzed the behavior of six different feature selection metrics, and observed that Odds Ratio (OR) and Document Frequency (DF) seem to perform cut-off of a greater number of features, if compared to Chi-Squared (CHI) and Information Gain (IG). [Almeida et al., 2011] evaluated the performance of different feature selection using Bayesian classifiers in a spam classification task, and noticed that Chi-Square test, Document Frequency and Information Gain offered the greatest removal of features while not losing in overall performance. [Basu and Murthy, 2012] proposed a new score called Term Significance (TS) to be applied as a feature selection metric, and results demonstrated that the overall classification performance using TS outperformed the classification applying other metrics, such as Information Gain, Document Frequency and Chi-Square test.

Neverthless, in reviewing studies conducted to evaluate the performance of several feature selection metrics, it is not clear which is the most recommended metric for text classification problems in general. Each classification task might yield different results when various metrics are applied. Therefore, a reasonable strategy to define the most suitable feature selection metric is to take into account the characteristics of different metrics with regards to the classification problem addressed. In the context of this research, the Odds Ratio (OR) and Inverse Document Frequency (IDF) were used as feature selection metrics. Because the dataset applied in our experimental settings is annotated with specific domain annotations, grouped by given categories (which are relevant bioentities, further explained in Section 3.4.1), the Odds Ratio metric is then useful to evaluate the discriminative power of each domain category to predict relevant documents. In addition, as the task handles a large document collection, the Inverse Document Frequency is a useful metric to take into account when evaluating the utility of a given term. Below we further explain these two feature selection metrics.

2.3.1 Inverse Document Frequency (IDF)

Considered one of the most simple feature selection measures, the Document Frequency represents the number of documents in the collection in which a term occurs. However, if we consider Zipf's law [Zipf, 1932] on the distribution of words, a term that has both a high term frequency and a high document frequency, (which is the case with the so-called stop-words, such as *the, and, are, is*) might not be very discriminative, since this term appears too frequently across all documents, and probably is far too common to contribute with a significant decision power to the model. To alleviate this, a well known score called *idf* utilizes the Inverse Document Frequency (IDF) [Sparck Jones,

1972]. The idf_t weight indicates if a term t is less frequent in the document collection, which is evidence of a better discriminative power.

The Inverse Document Frequency (idf_t) of a feature t in a document collection D is computed as described below:

$$idf_t = log\frac{N}{df_t} \tag{2}$$

where N is the number of documents in the dataset and df_t represents the number of documents that contain the term t.

2.3.2 Odds Ratio (OR)

The Odds Ratio (OR), described in [Szumilas, 2010], is a measure of the odds of observing an outcome given that a particular variable was seen, compared to the odds of observing the same outcome given that this variable was absent. In a binary classification scenario with a class A and a class B, the Odds Ratio calculates if the chances of encountering a feature are related to the fact that class A was also encountered, divided by the chances of not encountering the same feature given the fact that class A was encountered.

The Odds Ratio OR of a term t given a class C can be computed as follows:

$$OR_{(t,C)} = \frac{\frac{n_{Ct}}{n_C} / \frac{n_{C\bar{t}}}{n_C}}{\frac{n_{\overline{C}t}}{n_{\overline{C}}} / \frac{n_{\overline{C}\overline{t}}}{n_{\overline{C}}}} \tag{3}$$

where

n_{Ct} is the number of times the term t was seen in class C,

n_C the number of documents in class C,

$n_{C\bar{t}}$ is the number of documents in class C that do not contain term t,

$n_{\overline{C}t}$ is the number of documents not in class C that contain term t,

$n_{\overline{C}\overline{t}}$ is the number of documents with neither class C nor term t and

$n_{\overline{C}}$ is the number of documents not in class C.

2.4 Classifiers and Imbalanced Data Approaches

In this section, a brief survey of classification algorithms applied in combination with sampling and feature selection techniques is presented.

Several studies have evaluated the use of Support Vector Machines (SVM) [Vapnik, 1995] to handle the imbalance issue and described it as a sensitive algorithm to imbalanced corpora. [Akbani

et al., 2004] described the technique SMOTE with Different Costs (SDC), a combination of SVM and an oversampling method. The system yielded better performance when compared to a standard SVM with undersampling methods. However, the study specified that in their experiments, the SDC algorithm was based on the assumption that positive instances are somehow related, and therefore stand next to one another in the dimensional space. These conditions, however, should not be considered typical, since it is not guaranteed that in an imbalanced scenario, all minority instances are similar in content.

[Tang et al., 2009] presented a model called Granular SVM (GSVM), that included the use of an undersampling method. In general, this model demonstrated better performance compared to a standard SVM implementation or the independent use of an undersampling method. However [Tang et al., 2005] described that the GSVM model was likely to over-fit the training data, since the sampling process is repeated for several iterations until only the considered most informative majority instances are kept. The overfitting aspect is a drawback that can affect the classification of unseen instances.

[Mountassir et al., 2012] analyzed the performance of three algorithms under the implementation of varied undersampling methods. After experimenting with SVM, Naïve Bayes and k-Nearest Neighbor (k-NN) classifiers, the results showed that SVM was the most sensitive algorithm to handle data with imbalanced class distributions. All the undersampling methods evaluated seemed to perform in a very similar manner on the dataset that presented the most imbalanced distribution, where the rare class was represented by only 8% of the instances.

To handle severely imbalanced data in a text classification challenge, [Charton et al., 2013] described an approach that combined the use of feature selection techniques and the Logistic Model Trees (LMT) algorithm, which was defined in [Landwehr et al., 2005]. Handling a multi-class corpus where the minority classes represented $\approx 8\%$ and even $\approx 0.6\%$ of the entire collection, this approach managed to outperform all systems participating in the challenge. The results also demonstrated that the LMT algorithm outperformed the other classifiers, such as Naïve Bayes, Decision Trees, and even SVM, that was previously shown as a suitable algorithm to deal with imbalanced data.

In this chapter we have presented an overview of previous work conducted in biomedical text classification, as well as in tasks handling imbalanced datasets. We described well-known techniques used for feature selection and reviewed classification algorithms usually applied for imbalanced learning. In Chapter 3, we will provide a detailed description of the methodology used for conducting our experiments, including the dataset generation, document representation, extraction and selection of features, as well as a description of the classification algorithms used.

Chapter 3

Experiment Methodology

The task of biomedical text classification usually implies the processing of a large document collection, as well as the difficulty of having to identify a relatively small subset of the document instances to be categorized as meaningful for the task. In Chapter 2, a review of the state-of-the-art approaches utilized in tasks similar to the biomedical literature triage demonstrated that techniques to tackle the imbalanced class distribution and sparse feature space would be suitable to improve the performance of classification algorithms.

In an attempt to address these problems, the approach described in this Chapter evaluates the combination of:

- sampling techniques, in order to handle the imbalance issue, and
- feature selection techniques, in order to manage the feature space size.

Both techniques are introduced in the classification pipeline as pre-processing steps to the learning phase. The goal of adopting these techniques is to improve the algorithm performance when predicting the output of document instances belonging to the minority class, less represented in the collection, and therefore more challenging to be correctly classified by the algorithm.

As discussed in Chapter 2, undersampling methods, described in [Loyola-González et al., 2013] [Rahman and Davis, 2013], and feature filtering methods, described in [Wasikowski and Chen, 2010], have been previously shown to be beneficial when applied in classification tasks that presented issues similar to the ones found in the biomedical literature triage. In our work, these methods are used to design several classification models, and determine the most fitting approach to perform the task.

In this chapter the methodology adopted to conduct the set of experiments is explained. Sections 3.1 and 3.2 present the generation and composition of the dataset. Section 3.3 explains the progressive application of undersampling factors in the collection. Section 3.4 presents the extraction

and selection of domain-related features and the creation of a feature space formed by document feature vectors; while Section 3.5 provides a brief description of the classification algorithms used in the model design.

3.1 Dataset

The experiments described in this thesis were executed based on the *mycoSet* dataset, which was created in order to reproduce the scenario of the literature triage task. In particular, *mycoSet* contains scientific abstracts that compose a document collection which replicates the manual curation task for the mycoCLAP fungal genes database, described in Murphy et al. [2011]. All documents in *mycoSet* are abstracts of scientific papers, collected after querying the PubMed scientific database[1]. Specific queries were created by curators to retrieve a set of documents that are potentially relevant candidates for biomedical literature triage, for a given research topic. The queries are composed by research-related strings and a certain time range. For example, to retrieve potential fungal enzyme related documents, a query can be composed by an enzyme name/family, a logical conjunction, and the generic string "fung", to match all related fungal documents, as in the examples:

```
[ peroxidase AND fung* ] [ glucose oxidase AND fung* ]
```

mycoSet document collection was gathered after querying PubMed with variations of the example query described above, using 45 different enzyme names/families, and with a publication date up to December 31st, 2013. All 45 queries used to compose the *mycoSet* corpus are listed in the Appendix A of this thesis. *mycoSet* contains a total of 7,583 document instances related to fungal enzymes.

In order to support the manual curation of *mycoSet*, all documents were pre-processed with the mycoMINE text mining system [Meurs et al., 2012]. mycoMINE was used to annotate relevant units of text that represent bioentities. All entities annotated by mycoMINE were defined by biocurators as potential indicators of finding information on fungal enzymes. After being annotated with relevant entities, these documents were manually labeled by biocurators, who classified instances as being potentially relevant for further curation or not relevant to provide new data to their research. Documents considered as relevant were labeled *positive*, and therefore will be retained for full curation. On the contrary, documents considered as not relevant were labeled *negative*, and rejected by curators.

The manual curation effort performed on the document collection resulted in the identification of 749 positive documents, while 6,834 were rejected. All PubMed IDs of the documents used to compose *mycoSet* corpus and their assigned label are available as described in the Section 5.2 of

[1] http://www.ncbi.nlm.nih.gov/pubmed

this thesis. The number of positive documents compared to the number of negative documents in *mycoSet* demonstrates the highly imbalanced characteristic of the dataset. The majority class, corresponding to the documents labeled as non-relevant, represents 90.12% of the total number of instances in the corpus, while the minority class, corresponding to the documents labeled as relevant, is represented by only 9.88% of all instances. Table 3.1 summarizes this information and other statistics on *mycoSet*. In addition to the imbalanced class distribution, Table 3.1 also shows how sparse the feature space can be, since over 50,000 annotations were found in the document abstracts and titles.

Attribute	**Number**	**%**
Total number of instances	7,583	100%
Total number of instances with text content	6,898	90.96%
Negative instances	6,834	90.12%
Positive instances	749	9.88%
Number of words in paper abstracts	43,598	-
Number of words in paper titles	12,388	-
Number of annotations in paper abstracts	50,866	-
Number of annotations in paper titles	8,172	-
Number of Enzyme Comission (EC) numbers	12,272	-

Table 2: Statistics on the *mycoSet* corpus

3.2 Training and Test Corpora

Document instances in the *mycoSet* collection were separated into training and test data to perform supervised learning. The training data is used by the algorithm to learn a function that allows it to generalize a model from these instances, making it capable of predicting the class output of new documents; in this case, predicting if a document is relevant or not for the triage task. After learning a function with the training data, the test data, which contains only unseen instances, is then used to evaluate the performance of the classification model.

To generate the test data from the *mycoSet* corpus, a portion equivalent to 20.5% of all document instances was chosen randomly. The random selection of relevant and non-relevant instances to compose the test data was applied so that the class distribution would be similar to the realistic ratio of relevant documents encountered by curators when performing biomedical literature triage. Hence, the test set contains ≈10% positive instances and ≈90% negative instances. It is pertinent to evaluate the model performance on a pragmatic scenario, since it can be later introduced in the curators routine in order to predict the class output of new document instances, as an actual tool to support the triage task.

Later, once the test data was isolated, the remaining document instances were used to generate the *mycoSet* training data. At this point, about ≈6,000 instances were left. This document collection is not only large for a human biocurator to evaluate, but also presents the imbalanced class distribution that biocurators have to handle in the manual triage task. This large and uneven group of remaining documents was then utilized to generate several training corpora through sampling methods. As specified in Section 2.2.2, this technique was used to tackle the imbalance issue and to handle the great number of instances in *mycoSet*. The random sampling method employed in the experimental settings is further explained in Section 3.3.

3.3 Corpus Sampling

Since the *mycoSet* training data contains an imbalanced distribution between relevant and not relevant instances, a sampling technique is used. As discussed in Section 2.2.2, undersampling methods have been applied in order to handle these issues in classification tasks of biomedical datasets (e.g., [Tang and Zhang, 2006], [Zhang et al., 2011], [Varassin et al., 2013], [Yu et al., 2013]).

Undersampling was utilized to generate several training corpora from the non-test set instances of *mycoSet*. To evaluate different classification models, a random undersampling strategy was applied to gradually discard a percentage of the majority (not relevant) documents from the instances left in *mycoSet*. The progressive undersampling strategy applied is shown in Figure 2. This undersampling of the *mycoSet* majority training instances allows the generation and comparison between various models, each one of them with a different bias severity.

The algorithm developed to perform the *mycoSet* test set sampling is shown in Figure 3, and the algorithm developed to generate the *mycoSet* training sets undersampling is shown in Figure 4. The data sampling pipeline starts with the Algorithm 1, selecting randomly the given ratio of instances from the original list of *mycoSet* positive and negative documents. These randomly selected instances were then isolated to compose the test set, and not taken into account for the generation of training sets. Next in the pipeline, Algorithm 2 is executed, by considering a given set of parameters, such as the number of training sets to be created, the initial size for the training sets, the sampling factor used to gradually balance the class distributions, as well as the current list of available positive and negative document instances, after the test instances were isolated. Using the undersampling approach described by these algorithms, 9 training sets were generated by randomly selecting available document instances, with different class distributions. Table 3 shows the various USF applied in the majority class and the number of instances per class at each training set.

The first training set generated holds a similar class distribution as the *mycoSet* dataset, that represents the real imbalanced scenario of the literature triage. This set, with a 0% undersampling

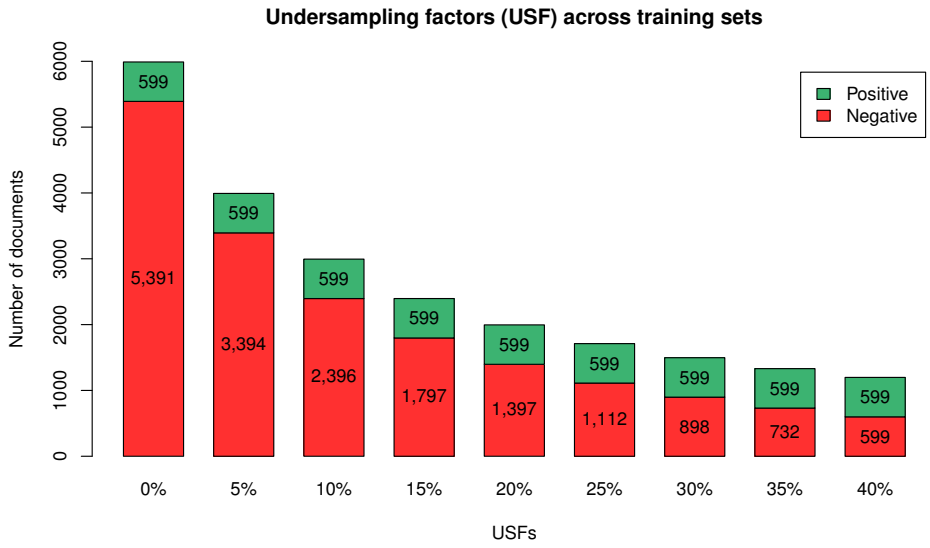

Figure 2: Balance of positive and negative instances in *mycoSet* training sets using undersampling

Undersampling factor	**Negative**	**Positive**
Undersampling 0%	5,391 (90%)	599 (10%)
Undersampling 5%	3,394 (85%)	599 (15%)
Undersampling 10%	2,396 (80%)	599 (20%)
Undersampling 15%	1,797 (75%)	599 (25%)
Undersampling 20%	1,397 (70%)	599 (30%)
Undersampling 25%	1,112 (65%)	599 (35%)
Undersampling 30%	898 (60%)	599 (40%)
Undersampling 35%	732 (55%)	599 (45%)
Undersampling 40%	599 (50%)	599 (50%)

Table 3: Percentage of instances across progressive undersampling of the *mycoSet* training data

Algorithm 1: Test set sampling

Require: Test set positive ratio TS_{PosRat},
a test set size TS_{Size},
a list of positive instances $PosList$,
a list of negative instances $NegList$;

compute test negative ratio as
$TS_{NegRat} = TS_{Size} - TS_{PosRat}$;

compute number of test positive instances as
$TS_{Pos} = (TS_{Size} * TS_{PosRat})/100$

compute number of test negative instances as
$TS_{Neg} = TS_{Size} - TS_{Pos}$

for all (instances in $PosList$) **do**
  $TS_{PosList} = TS_{Pos}$ selected randomly;
end for

for all (instances in $NegList$) **do**
  $TS_{NegList} = TS_{Neg}$ selected randomly;
end for
new test corpus $TS = TS_{PosList} + TS_{NegList}$;

Figure 3: Algorithm used to perform the *mycoSet* test set sampling

Algorithm 2: Training set undersampling

Require: Train initial positive ratio TR_{InPos},
a sampling factor S_{Fac},
an initial training set size TR_{InSize},
a number of training corpora TR_{Num},
a vector of positive ratios $|TR_{PosRat}|$,
a vector of training set sizes $|TR_{Sizes}|$,
a list of positive instances (without test instances) $PosList - TS_{PosList}$,
a list of negative instances (without test instances) $NegList - TS_{NegList}$;

for $(i = 0;\ i < TR_{Num};\ i++)$ **do**
compute training positive ratio $|TR_{PosRat_i}| = TR_{InPos} + (S_{Fac} * i)$;
end for

compute number of training positive instances as
$TR_{Pos} = (TR_{InSize} * TR_{InPos})/100$;

for $(i = 0;\ i < |TR_{PosRat}|;\ i++)$ **do**
compute the training set sizes as
$|TR_{Sizes_i}| = (TR_{Pos} * 100)/|TR_{PosRat_i}|$;
end for

for $(i = 0;\ i < |TR_{PosRat}|;\ i++)$ **do**
$TR_{NegRat} = (100 - |TR_{PosRat_i})$
compute number of training negative instances as
$TR_{Neg} = (TR_{NegRat} * TR_{Pos})/|TR_{PosRat_i}|$;

for all (instances in $PosList - TS_{PosList}$) **do**
$TR_{PosList} = TR_{Pos}$;
end for

for all (instances in $NegList - TS_{NegList}$) **do**
$TR_{NegList} = TR_{Neg}$ copied randomly;
end for

new training set $TR_i = TR_{NegList} + TR_{PosList}$;
end for

Figure 4: Algorithm used to perform the *mycoSet* training sets undersampling

factor (USF), is utilized to design a baseline model, as it contains the task natural ratio of relevant and non relevant documents. Next, new training sets were generated by gradually increasing the USF by 5%, until it reached a 40% USF of the majority instances, when the distribution between both classes became equal (i.e., 50% of positive instances and 50% of negative instances). Each training set was evaluated separately, providing a comprehensive comparison of class balances.

3.4 Document Representation

The data instances in *mycoSet* are represented in terms of features, so the classification function can be learned from the training set and then later evaluated on the test set. Document features are fragments of information identified in the text, that are used as an input to build the classification models. The extraction of these features was made on specific standard text fields, generally encountered across all documents: *AbstractText*, *ArticleTitle* and *RegistryNumber*. While the *AbstractText* and *ArticleTitle* hold the document abstract content and title, respectively, the *RegistryNumber* field contains Enzyme Commission (EC) numbers. EC numbers are numerical enzyme nomenclature used for identifying different chemical reactions catalyzed by enzymes [Webb, 1992]. Features are mostly composed by domain annotations, EC numbers and bag-of-words (BOW) representation of the domain annotations. In the next sections, the types of features considered for the experimental settings and the methods to perform feature extraction are presented.

3.4.1 Feature Extraction and Types

Document instances are submitted to a normalization step before the features are extracted. The normalization process is responsible for removing extra blank space in sentences, markup tags, as well as ASCII special characters, such as punctuation marks.

After normalization, the feature extraction process is executed on all documents to represent them by means of bioentities and domain annotations, provided by the mycoMINE system, as well as EC numbers. Domain annotations are grouped according to their context span in the text: an annotation can be at the sentence span, when the annotated content takes into account an entire sentence; or an annotation can be at the entity span, when it is usually composed by a single word or a short sequence of words.

The annotations encountered at an entity span are extracted and then kept as after the document normalization process, while annotations found at a sentence span are extracted after the normalization, but later represented in the feature space as a BOW.

In order to reduce the data sparseness of the feature space, after the sentence span features are expressed as a BOW, tokens that contain less than 3 characters or tokens that are found in

the PubMed stop-words [National Center for Biotechnology Information, 2005b] list are discarded. These words are not taken into account to build the classifier because their contribution to the discriminative power of the model is not worth the drawbacks that they represent in the entire process, such as the increase in the sparseness and in the learning time.

The 22 bioentities annotated by mycoMINE in *mycoSet* and their corresponding span are listed in Table 4. These entities were defined as the most significant by biocurators performing literature

Entity	**Span**	**Entity**	**Span**
AccessionNumber	entity	Glycosylation	sentence
ActivityAssayConditions	sentence	Kinetics	sentence
Assay	entity	Laccase	entity
Buffer	entity	Lipase	entity
Characterization	entity	Peroxidase	entity
Enzyme	entity	pH	sentence
Expression	sentence	ProductAnalysis	sentence
Family	entity	Temperature	sentence
Fungus	entity	SpecificActivity	sentence
Gene	entity	Substrate	entity
GlycosideHydrolase	entity	SubstrateSpecificity	sentence

Table 4: The 22 bioentities and spans annotated in *mycoSet* by the mycoMINE text mining system

triage of fungal enzymes. The following excerpt, taken from the *mycoSet* corpus, has been annotated with bioentities by mycoMINE.

<SubstrateSpecificity>The substrate specificity of three <Enzyme>ligninase </Enzyme>isozymes from the white-rot fungus <Fungus>Trametes versicolor</Fungus>has been investigated (...). </SubstrateSpecificity>(...) <RegistryNumber>EC 1.14.99.-</RegistryNumber>

To better exemplify the domain annotation spans and the document representation in terms of features, the information extracted from this *mycoSet* excerpt is listed below:

- Bioentities of the entity span: *[ligninase, Enzyme]; [Trametes versicolor, fungus].*
- Bioentities of the sentence span: *[substrate, substratespecificity]; [specificity, substratespecificity]; [three, substratespecificity]; [ligninase, substratespecificity]; [isozymes, substratespecificity]; [whiterot, substratespecificity]; [fungus, substratespecificity]; [trametes versicolor, substratespecificity]; [investigated, substratespecificity].*
- EC number list: *[11499].*

After extracting the domain annotations and EC numbers, the set of features is evaluated by a feature selection process. The methods applied are described in Section 3.4.2.

3.4.2 Feature Selection Strategy

The larger the set of features extracted, the larger and sparser is the dataset representation matrix. A sparse matrix reduces the accuracy of the classification models. Moreover, a large matrix can be costly in terms of computational processing during the training phase, because the greater the number of features in the matrix, the more time a classifier will take to fit a model for a given task. Techniques to reduce the feature space through feature selection can be valuable in such cases.

In this work, we explore a few standard feature selection methods in addition to sampling techniques. Before applying feature selection methods, the extracted features were narrowed by two initial criteria. First, the features were considered according to their frequency of occurrence, as an effort to maintain a more compact feature space, therefore words occurring less than 2 times in the training corpus were discarded. Second, features were considered according to their length, were all features with less than 3 characters were not taken into account when generating feature vectors.

As indicated in Chapter 2, in order to avoid data sparseness in the model and cut down the learning time of the classifier, two feature selection metrics were used to filter out attributes with a low score (indicating a potentially low discriminative value) and reduce the feature space size. Here we explain the strategy used to apply the Odds Ratio filtering and also the Inverse Document Frequency filtering, used as feature selection methods in the design of our models.

The feature selection using Odds Ratio as a metric was performed in the following manner: first, an odds ratio score was computed for each feature extracted by a model. Then, a confidence interval for each odds ratio score was computed, with a confidence level of 95%. At this point, two conditions were applied to perform the filtering. Features that presented:

1. a confidence interval that includes the null hypothesis (i.e., value of 1.0); or

2. an odds ratio that is less or equal to the null hypothesis (i.e., value of 1.0)

were discard from the feature set. The remaining features were further kept, in order to design the classification models.

The feature selection using Inverse Document Frequency as a metric was performed in the following manner: first, the Inverse Document Frequency of each feature was computed, considering its occurrence in both positive and negative classes. Then, similarly to the odds ratio filtering, all features with an Inverse Document Frequency score smaller than 1.0[2] were discarded.

In Section 4.4.3, the efficiency of each feature selection method will be shown and analyzed in the context of our biomedical literature triage task. The group of features selected by each metric used as feature selection among all extracted features in the *mycoSet* training data was utilized to construct

[2]This value was set intuitively, but has no theoretical foundations.

various feature vectors, applied in our classification models. The feature vector representation is explained in Section 3.4.3.

3.4.3 Feature Vector

The feature vector is a representation of document instances in the dataset in terms of feature occurrence. All documents in the *mycoSet* training and test sets are expressed as feature vectors, that will be later fed to the classifiers. One vector corresponds to a document, and each value in a vector corresponds to the number of times a certain feature was encountered in this document.

The *mycoSet* training and test sets are represented by a $F \times I$ matrix, where F is the number of features, and I is the number of document instances in the set. To illustrate such a matrix, after extracting features from the *mycoSet* excerpt presented in Section 3.4.1, its feature vector is represented as in Table 5. The columns represent the features extracted from the excerpt, while

ligninase	*Trametes versicolor*	*synthetic*	*substrate*	*specificity*	*three*	*fungus*	*enzyme*	...
2	2	1	1	1	1	2	1	...

Table 5: *mycoSet* sample feature vector representing feature occurrences for one document

the row holds the number of times each of these features was seen in this *mycoSet* excerpt. This numerical representation of each *mycoSet* document is submitted to a classification algorithm. The algorithms utilized in our experimental framework are described in Section 3.5.

3.5 Classification Algorithms

To evaluate the combination of data sampling and feature selection methods, the feature vectors are submitted to three different classification algorithms: Naïve Bayes (NB), Logistic Model Trees (LMT) and Support Vector Machine (SVM). The algorithm implementations were provided by WEKA [Hall et al., 2009]. WEKA is a machine learning workbech developed in Java, that offers general tools to perform data mining and automatic classification. For our experimental framework, the WEKA core implementations were used, with no posterior modifications to the algorithms parameters.

We understand that the three classification algorithms chosen to compose the models will provide us with different and interesting perspectives of approaching the triage task. First, the use of a NB classifier to evaluate the classification performance of our models provides a baseline perspective of the application of sampling and feature selection methods. Second, the use of a SVM classifier was previously reported in tasks that handled imbalanced data (e.g., [Mountassir et al., 2012], [Tang et al., 2005], [Tang et al., 2009]), thus it is reasonable to analyze if it is also effective to tackle the triage

classification. Third, the application of the LMT algorithm resulted in a noticeable performance in a classification task with highly imbalanced datasets, described by [Charton et al., 2013]. In light of this, and in addition to the fact that LMT can provide interesting points of comparison with models usually applied in imbalanced scenarios using SVM classifiers, the LMT algorithm was also considered for our experimental framework. Below, a brief review on each of these classification algorithms is given.

3.5.1 Naïve Bayes

A Naïve Bayes classifier is a probabilistic model based on Bayes' Rule, that assumes a strong conditional independence of features. This classifier builds a "naïve" independence model, considering that in a feature vector F, the features $F_1, ..., F_n$ are conditionally independent from each other, given a class C. By this assumption, Naïve Bayes implies that the presence of one word (one feature) is not correlated with the presence or absence of another word in a document, given a class label. Therefore, the probability of a document instance D (represented by a vector F) belonging to class C, $P(C|D)$, can be computed as:

$$P(C|D) = P(C|F) = P(C)\prod_{i=1}^{n} P(F_i|C) \tag{4}$$

where $P(C)$ is the prior probability of a class C, $P(F_i|C)$ is the discriminative value of a feature F_i found within a document D with regards to the class C, and n is the number of features. Naïve Bayes aims to identify the best $P(C|D)$, for all existing C. Hence, the classifier seeks to maximize a classification score for each document, as in:

$$class(D) = argmax P(C|D) = argmax P(C)\prod_{i=1}^{n} P(F_i|C) \tag{5}$$

where $class(D)$ is the class value that maximizes $P(C|D)$. This value is defined after the class prior probability $P(C)$ and each document feature value $P(F_i|C)$ are computed.

3.5.2 Logistic Model Tree

Logistic Model Tree consists of a combination of Decision Tree and LogitBoost algorithms. A Logistic Model Tree is a decision tree, with logistic regression models on its nodes. At each node of the decision tree, the LogitBoost algorithm is used to train a data subset for a certain number of iterations. This number is defined through five fold cross validation. An error rate is computed at each iteration, and the iteration presenting the lowest error rate is selected to define a logistic regression model for the current node. A Decision Tree criterion (e.g., maximum information gain) is then applied to split the current data subset. A LogitBoost execution to be started at the child

nodes will be initialized from the logistic regression model previously defined at the parent node. Tree splitting will be performed until there is still a relevant information gain.

In a Decision Tree model, leaves usually hold a class prediction as output. In a LMT model, leaves hold a logistic regression function for the current data subset at this node. A logistic function generated in a LMT leaf contains a substantial model: it does not only represent the data within the current node, but instead the logistic function has been continuously incremented, since it has been built on top of the same logistic function first defined at the root node. The final model of a Logistic Model Tree is defined as follows:

$$f(x) = \sum_{t \in T} (f_t(x) \cdot I(x \in S_t)) \tag{6}$$

where T represents the set of all leaves (terminal nodes), S_t is the dataset split on the current leaf t, $f_t(x)$ is the logistic regression function at the current node x. I is the indicator function: the expression $I(x \in S_t)$ has a binary evaluation, returning 1 only when the instance x belongs to the current dataset split S_t.

An example of a tree built by the LMT algorithm is shown in the Figure 5.

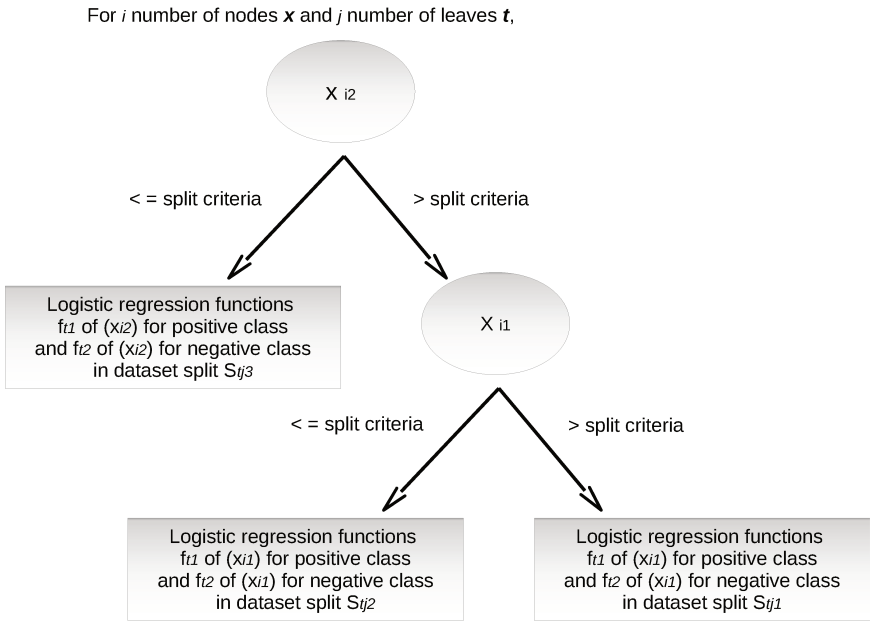

Figure 5: Overview of a Logistic Model Tree in a binary classification

3.5.3 Support Vector Machine

Support Vector Machine (SVM) is a well known algorithm that converges to an optimal solution for linear and non-linear classifications.

To separate data points on a dimensional space and tell their classes apart, a SVM computes the "margin maximum classifier" [Marsland, 2009]. A maximum margin is the largest radius around a classification boundary where no data points are placed. The closest data points encountered next to this margin are called support vectors. These vectors are considered as the hardest instances to be classified. Because of that, they are used as a "support" to draw a decision boundary and build a classification model.

If a classification problem is identified as linearly separable, the data points are simply separated by a line in the space. When linear separation is not possible, a SVM uses data transformation to separate the data point classes. The transformation computation is optimized to a linear decision with the use of a kernel function.

SVM classifies a new instance x according to its distance from the support vectors $\mathbf{x_i}$, and also from the hyperplane H, placed in the middle of a maximum margin. A weight vector $\vec{w}_i$ is placed orthogonally to the hyperplane, and the class prediction y_i for a new instance represents its coefficient on the weight vector. The decision function for SVM is computed as shown in the following equation:

$$f(x) = \sum y_i \vec{w}_i K(x, \mathbf{x_i}) \tag{7}$$

where y_i stands for the class prediction (+1 or -1 in a binary classification), $\vec{w}_i$ represents the weight vectors, K is the kernel function, x is the instance to be classified, and $\mathbf{x_i}$ represents the support vectors.

The hyperplane H drawn by the SVM algorithm can be seen in Figure 6. The hyperplane H^+ contains the support vectors of a positive class, while hyperplane H^- contains the support vectors of a negative class. A SVM classifier will compute H such that the distance between H^+ and H^- is maximized.

In this chapter we have presented the methodology used to design and perform our experiments. We described how the dataset was generated, the progressive application of undersampling factors in the training data and the framework applied to extract and select features. We also introduced the properties of the three algorithms applied in our model designs. In the next chapter we will discuss the evaluation of our system, explain the metrics used to assess the model performances and analyse the effect of undersampling and feature selection to handle the triage of biomedical literature.

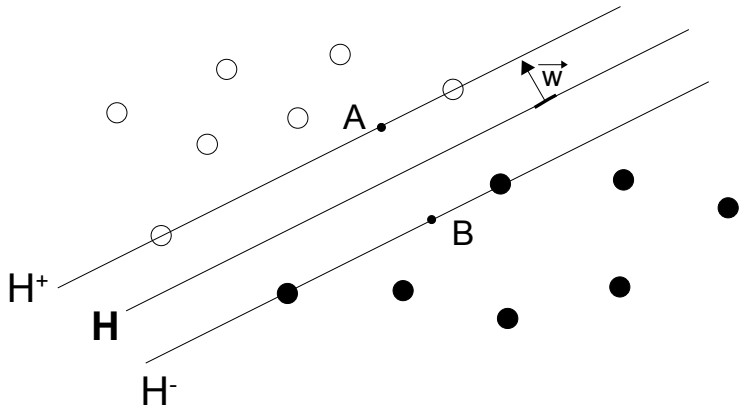

Figure 6: Separation of data performed by a SVM classifier given a hyperplane H

Chapter 4

System Evaluation

In this chapter we present and discuss the evaluation of our proposed system. Section 4.1 introduces the evaluation metrics used to assess and compare the performance of all 324 models. In Section 4.2 we describe in detail the experimental framework of each model by specifying the exact parameters used (the different feature sets, feature extraction and selection, classification algorithms and data undersampling factors). Then Section 4.3 presents the results obtained by all models. Next, in Section 4.4, we provide an overall analysis and discussion of the system scores.

4.1 Evaluation Metrics

The evaluation of classification algorithms can be made by several metrics. Usually one is interested in assessing if the classifier is able to achieve more correct predictions than incorrect predictions. To reflect the ratio of correct predictions made by a classifier, the classification accuracy is computed. The confusion matrix demonstrated in Table 6 outlines the predictions made by a classifier.

	Predicted Positive	**Predicted Negative**
Belong to the Positive class	True Positive (TP)	False Negative (FN)
Belong to the Negative class	False Positive (FP)	True Negative (TN)

Table 6: Confusion matrix of a binary classification task

The True Positive (TP) and True Negative (TN) are the document instances correctly classified in the positive and negative classes, respectively. False Negative (FN) are the positive instances classified as negative, while False Positive (FP) are the negative instances classified as positive. For the sake of simplicity, these same acronyms will be used for now on to represent the instances classification status and their numbers. To evaluate the accuracy *Acc* of a classifier, one should

compute the following:

$$Acc = \frac{TP + TN}{(TP + FN) + (FP + TN)} \tag{8}$$

where the sum of all correct predictions is divided by the total number of instances in the dataset. Most algorithms tend to maximize the accuracy during classification. However, this characteristic is very unfavorable for tasks that handle imbalanced data. At decision time, the class distribution in the dataset indicates to the classifier that the most expected output is the majority class. For example, in a context where ≈10% of the document instances belong to the minority class, the classifier can achieve an impressive accuracy simply by classifying all instances in the majority class, and ignoring all instances in the minority class.

However this is not a representative evaluation of the classifier performance in an imbalanced scenario, as demonstrated by previous studies (e.g., [Su and Hsiao, 2007] [He and Garcia, 2009]). If the minority class is the one that contains the most relevant instances for the classification task, evaluating a model performance by its accuracy is not a recommended approach since the actual interesting instances could be completely overlooked without affecting the performance scores [He and Garcia, 2009].

In order to properly evaluate the performance of the models on predicting the output of the minority instances, other metrics are usually employed. The F-measure [Makhoul et al., 1999] and the weighted F-β score are commonly applied to evaluate tasks that handle imbalanced datasets, as demonstrated by [Weiss, 2004], [He and Garcia, 2009], [Batuwita and Palade, 2009], [Ferri et al., 2009] and [Japkowicz and Shah, 2011]. These measures utilize the Precision and Recall of classifiers, which are generally computed in information retrieval and extraction tasks. In an imbalanced scenario, they are able to assess the classification performance on the minority class. Another related score that allows a less biased evaluation of the classifier is the Matthews Correlation Coefficient [Matthews, 1975], which has been listed as an alternative measure when handling imbalanced biomedical data [Baldi et al., 2000]. In line with the state-of-the-art, we therefore used these five other metrics to evaluate our models. These metrics are briefly explained hereafter.

Precision evaluates the proportion of correct predictions among correct and incorrect predictions that the classifier makes for a certain class. This measure indicates if a classifier is capable of outputting more relevant than irrelevant results. Precision is calculated by the True Positives instances (TP, i.e. correctly classified documents) divided by the sum of True Positives and False Positives instances (TP and FP, i.e. all class predictions).

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall represents the ratio of relevant predictions made by the classifier between all existing relevant instances that should have been predicted. This measure demonstrates the capability of a

classifier to predict the universe of relevant instances. Recall is calculated by the TP instances (i.e. correctly classified documents) divided by TP and FN instances (i.e. all instances belonging to the same class).

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Figure 7 shows a graphical representation of the distribution of TP, FP, FN and TN instances and their overlap with regards to the relevant and non relevant instances in a binary classification task. The circled area represents all predictions outputted by the classifier for a certain class.

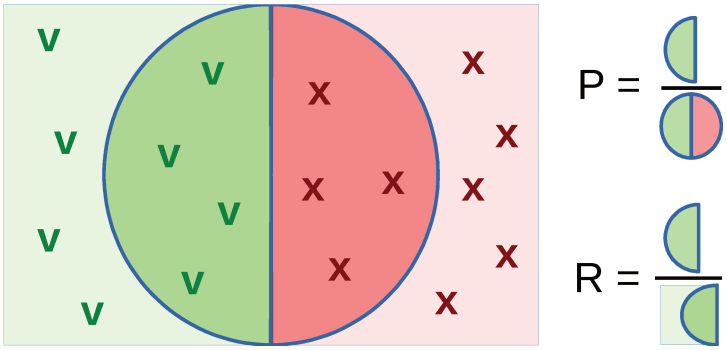

Figure 7: Computation of Precision (P) and Recall (R) using TP, FP and FN instances

The "v" symbols represent the universe of relevant instances, while the "x" symbols represent the universe of non relevant instances. The Precision of a class is calculated taking into account all instances included in the circled area. Recall is calculated using the instances found within the each square.

F-measure is a balanced, harmonic mean of Precision and Recall scores, obtained through the formula:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

F-$\boldsymbol{\beta}$ score is a generalization of the F-measure defined as follows:

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{12}$$

The value of $\boldsymbol{\beta}$ is the relative weight of Recall over Precision. Since in our experiments we are more interested in the model ability to identify the entire universe of relevant instances, Recall should be emphasized when calculating F-$\boldsymbol{\beta}$ score. Thus, the $\boldsymbol{\beta}$ value should be greater than 1. In our experiments, we used $\boldsymbol{\beta} = 2$, leading to the F-2 score.

Matthews Correlation Coefficient (MCC) represents a coefficient of agreement between observed and predicted classifications. A correlation value equal to 1 stands for a total agreement

(a perfect prediction), while a value equal to zero means total disagreement. MCC can be computed using the formula below:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

4.2 Experimental Setup

4.2.1 Set of Features

Extraction

The groups of features used across experiments in this work was derived from the feature extraction approach described in Section 3.4.1. The final types of features includes 5 groups:

F1: Annotated bio-entities

[e.g., *enzyme, fungus, substratespecificity*]

F2: Annotated contents of entity spans

[e.g., *ligninase, trametes versicolor*]

F3: Annotated contents of sentence spans (as a BOW)

[e.g., *substrate, specificity, three, ligninase, isozymes, whiterot, fungus...*]

F4: Enzyme Commission (EC) numbers (unique numerical identifiers for enzyme reactions)

[e.g., *11499*]

F5: BOW representation of the entire fields (ArticleTitle and AbstractText)

[e.g., *substrate, specificity, three, ligninase, isozymes, white, rot, fungus, trametes...*]

These groups of features were combined in different ways to perform our experiments. Additionally, we evaluated the performance of types of features on their own, without utilizing a combination of items. In order to identify the different combinations or types of features applied, we list here the 4 different sets of features considered in our experimental setting:

S1: The set of features S1 is formed only by [F1].

S2: The set of features S2 is composed by [F1 + F4].

S3: Set of features S3 is composed only by [F5].

S4: And finally, the set S4 is a combination of [F1 + F2 + F3 + F4].

All sets of features were evaluated across all classifiers and USFs, as well as feature selection metrics. Each set of features utilized will provide us with different perspectives of approaching the triage task. First, S1 will allow us to evaluate the discriminative power of the 22 bioentities identified by biocurators and the S1 based models capability to rely only in a small list of domain-oriented attributes. Second, the set S2 will provide an idea of the relevance of adding EC numbers in the list of attributes, when we compare the S2 based models with the S1 based models. Third, S3 provides us with a baseline, as the extraction approach is the generic approach to any text classification task. Finally, S4 will allow us to evaluate the discriminative power of the domain annotations in their entirely, since the bioentities and their annotated content are together taken into account.

Selection

After extracing from the training data the set of features chosen for a classification model, we used the feature selection strategies described in Chapter 3 to filter out the features with the lowest scores, according to the criteria and thresholds explained in Section 3.4.2.

Odds Ratio and Inverse Document Frequency filtering were applied separately in the design of the classification models. Both metrics were evaluated with the use of all 4 sets of features. This approach will allow us to better understand of the impact of feature selection methods considering the different types of features used in the task. This strategy will also provide us with the means to evaluate the performance of one feature selection metric compared to the other.

4.2.2 Classifiers

The classifiers used in our experiments are built-in algorithm implementations available within the Weka workbench [Hall et al., 2009]. The three classification algorithms previously described were utilized:

1. Naïve Bayes (NB)
2. Logistic Model Tree (LMT)
3. Support Vector Machine (SVM)

4.2.3 Undersampling

The undersampling technique was used to generate training corpora with different class distributions. As indicated in Section 3.3, a first training set was generated with a similar class distribution to the one characteristic of the biomedical literature triage.

After generating this first set, a progressive undersampling approach was applied to gradually discard negative instances in the corpus and generate several training sets, until the balance between positive and negative classes reached an equal distribution. A total of 9 training sets were generated, starting from a 0% undersampling factor (USF), where the set contains only 10% of positive instances, up to a 40% USF, where the training set has 50% of positive instances. All USFs and class distributions by percentage are listed below:

1. Training set with 0% USF: 90% negative, 10% positive
2. Training set with 5% USF: 85% negative, 15% positive
3. Training set with 10% USF: 80% negative, 20% positive
4. Training set with 15% USF: 75% negative, 25% positive
5. Training set with 20% USF: 70% negative, 30% positive
6. Training set with 25% USF: 65% negative, 35% positive
7. Training set with 30% USF: 60% negative, 40% positive
8. Training set with 35% USF: 55% negative, 45% positive
9. Training set with 40% USF: 50% negative, 50% positive

The progressive application of undersampling factors to generate the training corpora is also shown in Figure 2. Overall, the use of 4 sets of features, 3 classifiers, 9 USFs, 2 feature selection methods plus the use of no feature selection, lead to the evaluation of 4 x 3 x 9 x 3 = 324 models.

4.3 Experimental Results

We present here the performance scores in terms of precision, recall, MCC, F-measure and F-2 obtained after applying the 324 models designed for the triage of *mycoSet*. Since we are more interested in evaluating to which extent the models are capable of correctly classifying relevant instances, we will focus our analysis on the scores obtained for the positive instances. Therefore, all the results that we present in this section are the scores obtained only for the positive class. [1]

Figures 8 and 9 give an overview of our findings, by showing the two best classification models we identified after the evaluation of all different model designs. Figure 8 demonstrates the best model with no use of feature selection, when compared to the baseline approach. This model is composed by a LMT classifier, an equally balanced training set and the set of features S4 (all domain annotations).

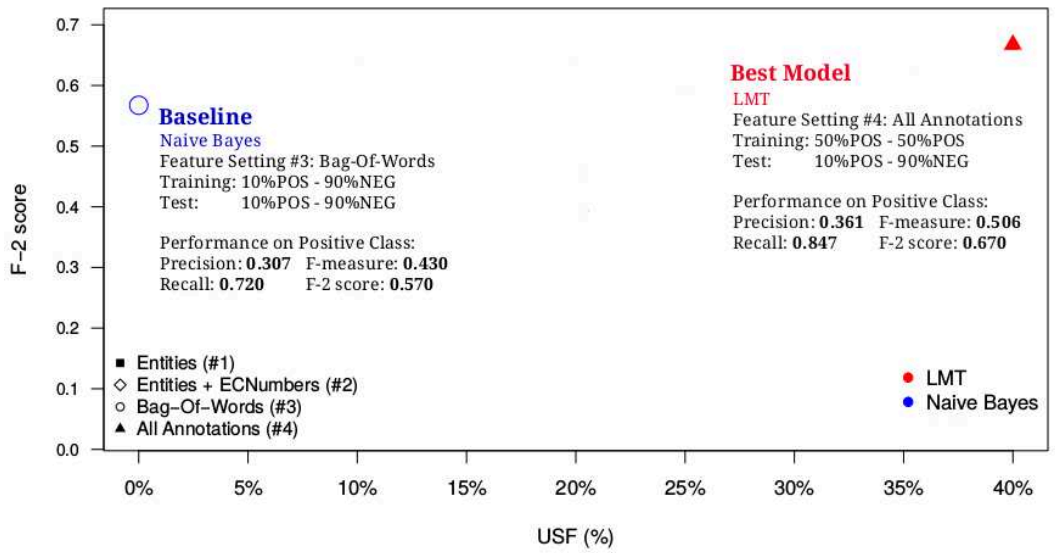

Figure 8: *mycoSORT* F-2 scores for the baseline and the best model for the approach using only USFs (TM1)

Figure 9 demonstrates the best model using Odds Ratio as feature selection, in comparison with the baseline approach. The settings of this model are also composed by an equally balanced training set and the LMT algorithm, with the difference that the features applied here were a subset chosen from the set of features S4 according to an Odds Ratio criteria.

We will now present the entirety of our experimental results with more details. The results obtained by our models are presented as a function of the set of features and strategy adopted. The complete tables are listed in the Appendix B of this thesis. Table 7 summarizes all our tables of results, the set of features and feature selection methods demonstrated in each one of them. Tables 18 to 21 (in Appendix B) present the scores for classification models using only the undersampling method as strategy. Tables 22 to 25 (in Appendix B) show the scores for the classification models using the undersampling method combined with Odds Ratio as a feature selection metric. Finally, Tables 26 to 29 present the scores for the classification models using the undersampling method combined with Inverse Document Frequency as a feature selection metric. The results reported in Tables 18, 22 and 26 show the performance achieved by using the set of features S1, which is composed of only the 22 bioentities, as explained in Section 3.4.1. The scores in Tables 19, 23 and 27 were obtained by using the set of features S2, composed of the 22 bioentities and all the EC numbers

[1]The results for the negative class overall vary around 90% to 98% for precision, 78% to 99% for recall and 84% to 95% for F-measure in models not using any feature selection method.

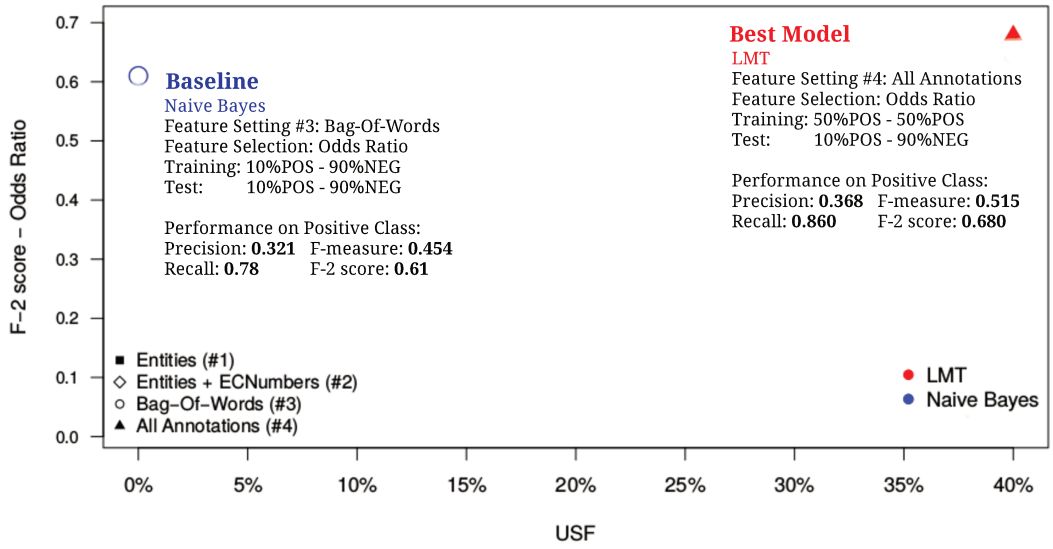

Figure 9: *mycoSORT* F-2 scores for the baseline and the best model for the approach using USFs and Odds Ratio (TM3)

Table	*Set of Features*	*Feature Selection*
18	S1	N/A
19	S2	N/A
20	S3	N/A
21	S4	N/A
22	S1	Odds Ratio
23	S2	Odds Ratio
24	S3	Odds Ratio
25	S4	Odds Ratio
26	S1	Inverse Document Frequency
27	S2	Inverse Document Frequency
28	S3	Inverse Document Frequency
29	S4	Inverse Document Frequency

Table 7: Summary of all of *mycoSORT* tables of experimental results

found in the documents. Results in Tables 20, 24 and 28 report the scores achieved by using the set of features S3, which contains the BOW representation of the article titles and abstracts. Tables 21, 25 and 29 show our results obtained by applying set of features S4, composed by all domain annotations, bioentities and EC numbers.

The combination of different USFs, classification algorithms, set of features and feature selection metrics derived a total of 324 models. Table 8 lists all charts utilized to show the performance of our classification models, the score being reported and the feature selection method applied in the models. Figures 10 to 13 summarize the scores presented in Tables 18 to 29. These charts

Figure	*Evaluation Metric*	*Feature Selection*
10	F-measure	N/A
11	F-2	N/A
15	F-measure	Odds Ratio
16	F-2	Odds Ratio
12	F-measure	Inverse Document Frequency
13	F-2	Inverse Document Frequency

Table 8: Summary of all of *mycoSORT* charts of experimental result

demonstrate which were the best performing classification algorithms and the most fitting set of features with regards to the USFs and feature selection approaches.

4.4 Discussion

We present here a discussion of the performances of our various classification models designed using the different USFs, sets of features, feature selection metrics and classification algorithms.

For the task addressed in our research, we considered as a baseline the classification model composed by the most naïve approach. Our baseline model uses a Naïve Bayes classifier, the set of features S3 (BOW) and the training set with 0% USF. This configuration constitutes our baseline because the parameters have not been tailored to our specific application and compose a generic approach to perform text classification. The baseline model for the positive class achieved the scores described in Table 9. These scores are also highlighted in Table 20. We will use the scores of this

Model	**Precision**	**Recall**	**F-measure**	**F-2**
Baseline	0.307	0.720	0.430	0.570

Table 9: *mycoSORT* results obtained for the baseline approach

baseline model as a reference to evaluate the results obtained by the other models designed and

tested in our experiments, and to measure the contribution of each parameter.

4.4.1 Most Discriminative Features

In Figures 10 and 11 we can observe the F-measure and F-2 scores for the models that applied only undersampling (no feature selection method).

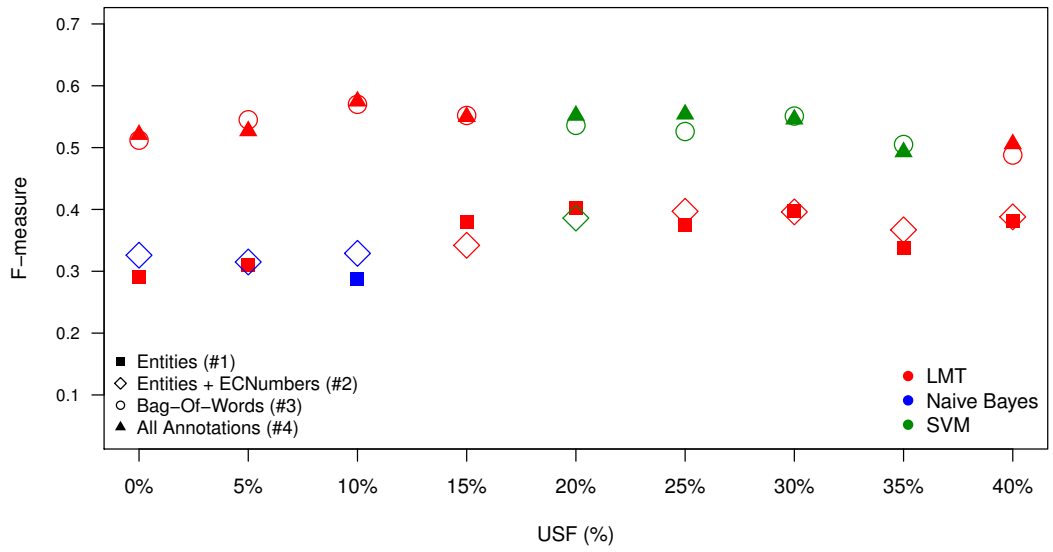

Figure 10: Summary of *mycoSORT* F-measure for the positive class. Best classifiers and set of features for each USF

We start by discussing the F-2 scores achieved with the four different sets of features, presented in Figure 11. A first observation is that the models designed with the set of features S4 (all domain annotations - shown by triangles), present the best performance overall considering the F-2 score obtained across the various USFs used in our experiments. The scores obtained with models using the set S4, in general, outperform the scores of models based on the set of features S3 (BOW - shown by circles), which is considered the baseline approach for feature extraction. Table 10 demonstrates a comparison between the scores obtained by the models using the set S3 and the scores of the models using S4. To summarize the comparison of models, we list here only the scores for the lowest (0%) and the highest USFs (40%). All scores in which the set S4 based models outperform S3 based models are highlighted. The complete results obtained by sets S3 and S4, across all USFs, can be found in tables 20 and 21.

Not only did the S4 based models outperform S3 based models, but they also have a feature

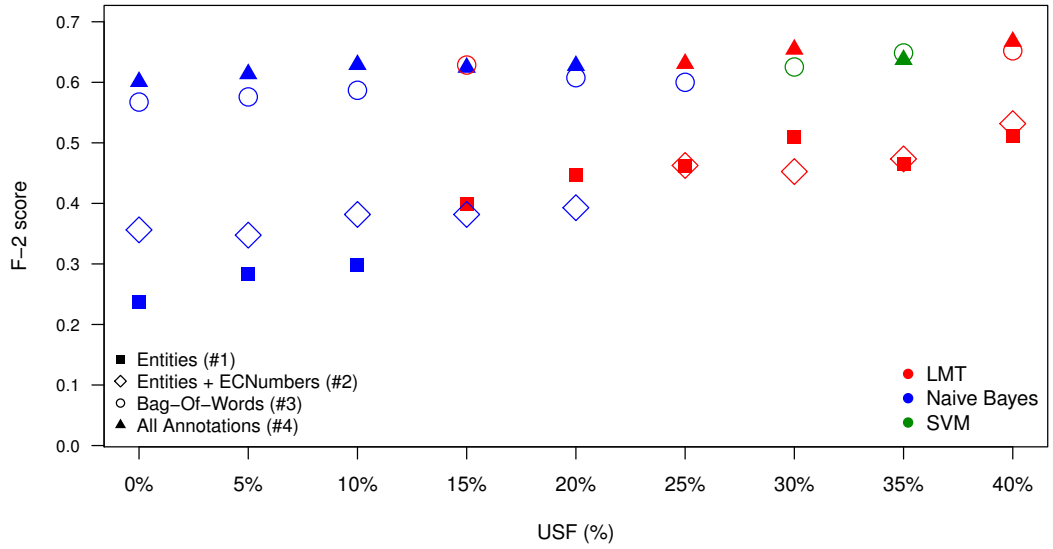

Figure 11: Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF

Setting	**Classifier**	**Precision**	**Recall**	**F-measure**	**F-2**
Set S3 (USF 0%)	Naïve Bayes	0.307	0.720	0.430	0.570
Set S4 (USF 0%)	Naïve Bayes	**0.355**	**0.727**	**0.477**	**0.600**
Set S3 (USF 0%)	LMT	0.656	0.420	0.512	0.450
Set S4 (USF 0%)	LMT	**0.685**	0.420	**0.521**	**0.460**
Set S3 (USF 0%)	SVM	0.833	0.033	0.064	0.040
Set S4 (USF 0%)	SVM	**0.867**	**0.087**	**0.158**	**0.110**
Set S3 (USF 40%)	Naïve Bayes	0.303	0.773	0.435	0.590
Set S4 (USF 40%)	Naïve Bayes	0.295	**0.780**	0.428	0.590
Set S3 (USF 40%)	LMT	0.344	0.840	0.488	0.650
Set S4 (USF 40%)	LMT	**0.361**	**0.847**	**0.506**	**0.670**
Set S3 (USF 40%)	SVM	0.338	0.840	0.482	0.650
Set S4 (USF 40%)	SVM	0.331	0.793	0.468	0.620

Table 10: Comparison of the positive class scores for models using sets of features S3 and S4 for the lowest and highest USFs

space ≈56% smaller. The feature space sizes are discussed in detail in Section 4.4.3. For instance, the feature space size of set S3 varies from about 7,622 features, when using 40% USF, to 20,729 features, when using 0% USF; while the feature space of set S4 varies between 3,338, with 40% USF, and 8,931, with 0% USF. The fact that the performance of S4 based models were better than S3 based models even with less than half the number of features indicates that the domain annotations used in S4 carry a higher discriminative power, when compared to the naïve BOW approach used in S3.

If we turn our attention to the scores obtained by set of features S1, we can observe another indication that the domain-related features carry an interesting discriminative capability. Even though the set S1 is composed of only 22 features (bioentities - shown by squares in Figure 11), the S1 based models achieved a reasonable performance in terms of F-2 score if compared to the models using the baseline features (S3), which have a considerably larger feature space, since it consists of the BOW representation of the document instances (over 7,600 features at 40% and over 20,000 features at 0% USF).

Setting	**Classifier**	**Precision**	**Recall**	**F-measure**	**F-2**
Set S1 (USF 40%)	Naïve Bayes	0.254	0.467	0.329	0.400
Set S3 (USF 40%)	Naïve Bayes	0.303	0.773	0.435	0.590
Set S1 (USF 40%)	LMT	0.269	0.660	0.382	0.510
Set S3 (USF 40%)	LMT	0.344	0.840	0.488	0.650
Set S1 (USF 40%)	SVM	0.196	0.667	0.303	0.450
Set S3 (USF 40%)	SVM	0.338	0.840	0.482	0.650

Table 11: Comparison of the positive class scores for models using sets of features S1 and S3 for the highest USFs

A comparison between the results obtained with models using S1 and S3 at 40% USF is shown in Table 11. Looking at the F-2 scores of these models, we can observe that the F-2 yield by S3 based models is ≈0.20 higher for the Naïve Bayes and SVM, while for the LMT the F-2 score is ≈0.14 higher, even though the set S1 is over 99% smaller than S3 in number of features. Approaches relying on the set of features S1 can be suitable in circumstances in which the computational cost, in terms of the time taken to fit a model and complete the learning phase, are important concerns.

Finally, a comparison between scores in Tables 18 and 19 indicates that the incorporation of EC numbers in the set of features S2 resultes in performance improvement for the models generated with the lowest and the highest USFs when compared to the results of S1 based models.

Table 12 summarizes the comparison of scores between S1 and S2 based models, demonstrating when S2 based models outperformed S1 based models. The scores that demonstrated improvement are highlighted. For these configurations, Naïve Bayes and LMT were the classifiers that best

Setting	Classifier	Precision	Recall	F-measure	F-2
Set S1 (USF 0%)	Naïve Bayes	0.286	0.227	0.253	0.240
Set S2 (USF 0%)	Naïve Bayes	0.285	**0.380**	**0.326**	**0.360**
Set S1 (USF 0%)	LMT	0.492	0.207	0.291	0.230
Set S2 (USF 0%)	LMT	**0.516**	0.107	0.177	0.130
Set S1 (USF 40%)	Naïve Bayes	0.254	0.467	0.329	0.400
Set S2 (USF 40%)	Naïve Bayes	0.244	**0.520**	**0.332**	**0.420**
Set S1 (USF 40%)	LMT	0.269	0.660	0.382	0.510
Set S2 (USF 40%)	LMT	0.267	**0.707**	**0.388**	**0.530**

Table 12: Comparison of the positive class scores for models using sets of features S1 and S2 for the lowest and highest USFs

performed. Again, to summarize our comparison, we show their scores for the lowest (0%) and highest (40%) USFs, and the complete results for all USFs can be seen in Tables 18 and 19. Using a less balanced training set, recall raises from 0.227 to 0.380, F-measure from 0.291 to 0.326 and F-2 score from 0.230 to 0.360 in the Naïve Bayes model. For the more balanced training set, recall raises from 0.660 to 0.707, F-measure from 0.382 to 0.388 and F-2 from 0.510 to 0.530, for the LMT model. This comparison can still be seen in Figure 11, where S2 based models (bioentities and EC numbers - shown by a diamond) outperform S1 based models at 0-10% USF and at 40% USF. Considering the scores obtained by the use of domain annotations in set S4 and bioentities in set S1 compared to the scores yield by using baseline features (S3), we are led to assume that the use of domain-related features are discriminative for this task. The domain-related annotations are best represented in the set S4, which we therefore consider as the most suitable set for our context, among the four sets evaluated.

4.4.2 Imbalanced Learning Strategy

Now we turn our attention to the strategy specially used to handle the imbalanced class distribution in the dataset. Relevant results are shown in Figures 10 and 11. First, we look at the fact that the use of an undersampling method entailed a reduction of 80% of the size of the training set, when compared to the training set containing the real triage imbalanced ratio. The training set composed by 90% of negative and 10% positive instances contains a total of 5,990 documents, while the 50% positive and 50% negative training set contains only 1,198 documents. Cutting down the size of the corpus used to learn the classification model is advantageous in our context, not only to handle the imbalance problem, but also to decrease the computational cost since less data is being processed, and therefore less time is required to complete the learning phase.

An observation we can draw from the scores in Figure 11 is that the use of a progressive undersampling strategy to generate the training sets resulted in a gradual improvement in performance.

The increase of F-2 scores follows the curve of progressive USFs applied in the training data. The models using higher USFs yield better scores than the models with low USFs, which used more imbalanced training sets. Models with high USFs also require less computational resources to be processed.

To complete our imbalanced learning strategy, we now evaluate the performance of the three classification algorithms, with regards to the progressive undersampling applied in the training sets. First, we analyze the F-measures shown in Figure 10. For the majority of the classification models across all USFs, the LMT algorithm (shown by the color red) outperformed the other classifiers in terms of F-measure. If we analyse Figure 11, we can observe that LMT outperformed the other two algorithms when the model is composed by training sets generated with higher USFs. We understand that this indicates that LMT provides better recall of positive instances compared to NB and SVM classifiers in scenarios where the dataset contains a more balanced class distribution.

Setting	**Classifier**	**Precision**	**Recall**	**F-measure**	**F-2**
Set S1 (USF 0%)	LMT	0.492	0.207	0.291	0.230
Set S1 (USF 40%)	LMT	0.269	**0.660**	**0.382**	**0.510**
Set S2 (USF 0%)	LMT	0.516	0.107	0.177	0.130
Set S2 (USF 40%)	LMT	0.267	**0.707**	**0.388**	**0.530**
Set S3 (USF 0%)	LMT	0.656	0.420	0.512	0.450
Set S3 (USF 40%)	LMT	0.344	**0.840**	0.488	**0.650**
Set S4 (USF 0%)	LMT	0.685	0.420	0.521	0.460
Set S4 (USF 40%)	LMT	0.361	**0.847**	0.506	**0.670**

Table 13: Comparison of the positive class scores for models using 0% USF and 40% USF with the LMT algorithm

To demonstrate the difference of performance between the lowest and the highest USFs, we compare the scores achieved by the LMT classifier across the four sets of features in Table 13, in which the scores that demonstrated improvement are highlighted. The recall of more balanced training sets is greater than in imbalanced models. For instance, in the S2 based model, the recall is improved by $\approx$0.6, while in S3 and S4 based models it improves $\approx$0.42. Therefore, the use of an equally balanced corpus combined with the LMT algorithm appears to produce the most fitting approach to handle the problem of imbalanced class distribution.

To illustrate the improvement gained by using the best strategies identified so far, we provide in Figure 8 a comparison between the baseline model and the best model identified after experimenting with different USFs, sets of features and algorithms. This comparison summarizes our conclusions so far, by demonstrating that the use of domain annotations (set of feature S4), with no feature selection, combined with an equally balanced training set (40% USF) and the use of the LMT

algorithm provides an improvement of ≈17% in recall and in F-2 when compared to a naïve approach. We call this first best model Triage Model 1 (TM1). The results obtained by the TM1 model are highlighted in Table 21.

4.4.3 Best Feature Selection Method

To evaluate if we could improve the model performance further, we studied the impact of applying Odds Ratio and Inverse Document Frequency as feature selection methods to filter out less discriminative attributes on the different sets of features. We expect that the feature selection methods will first reduce the amount of noise introduced in the models by non-discriminative features and possibly reduce overfitting; and second, reduce the computational cost of the learning phase and fit a model in less time than if the entire set of features was applied. Table 14 illustrates the impact of each feature selection method in the feature space size compared to the models with no feature selection. To provide an overview of the feature space size for the subset generated by Inverse Document Frequency and Odds Ratio feature selection, we demonstrate the reduction of the feature space size only for the most imbalanced (0% USF) and the most balanced (40% USF) models across all sets of features (S1 to S4).

Model	**# features**	**IDF**	**Reduction**	**OR**	**Reduction**
S1 with 0% USF	22	15	31.82%	17	22.73%
S1 with 40% USF	22	7	68.18%	14	36.36%
S2 with 0% USF	397	222	44.08%	58	85.39%
S2 with 40% USF	186	65	65.05%	32	82.80%
S3 with 0% USF	20,729	15,193	26.71%	2,073	90.00%
S3 with 40% USF	7,622	3,800	50.14%	908	88.09%
S4 with 0% USF	8,931	8,858	0.82%	1,564	82.49%
S4 with 40% USF	3,338	3,291	1.41%	681	79.60%

Table 14: Number of features in the models before and after applying feature selection methods

As shown in Table 14, the models using Inverse Document Frequency filtering demonstrated a greater variation in the reduction among the different sets of features. For example, for the set S1 the use of Inverse Document Frequency reduces the size of the feature space from 22 bioentities down to 7-15 features. For set S2 Inverse Document Frequency reduces the number of features by ≈44% in more imbalanced models (from 397 to 222 features) and to 65% in more balanced models (from 186 to 65 features). With set S3, the reduction was from ≈26.7% up to 50% for more balanced models. Finally, for set S4, the number of features is cut down only from ≈0.8% to 1.4%.

Contrarily to our expectations, Inverse Document Frequency filtering seems to have a negative impact on F-measure and F-2 scores compared to the results obtained with no feature selection

methods. We provide a comparison between the scores obtained by the models using no feature selection and models using Inverse Document Frequency in Table 15. These models are composed of the set of features S4 and the LMT classifier. To summarize our comparison, we show here the results of models using 0% USF and 40% USFs, and the complete results obtained by all USFs can be seen in Tables 21 and 29.

Setting	**Classifier**	**Feature Selection**	**Precision**	**Recall**	**F-measure**	**F-2**
Set S4 (USF 0%)	Naïve Bayes	N/A	**0.355**	**0.727**	**0.477**	**0.600**
Set S4 (USF 0%)	Naïve Bayes	IDF	0.350	0.720	0.471	0.590
Set S4 (USF 0%)	LMT	N/A	0.685	0.420	0.521	0.460
Set S4 (USF 0%)	LMT	IDF	0.717	0.440	0.545	0.480
Set S4 (USF 0%)	SVM	N/A	**0.867**	**0.087**	**0.158**	**0.110**
Set S4 (USF 0%)	SVM	IDF	0.750	0.020	0.039	0.02
Set S4 (USF 40%)	Naïve Bayes	N/A	**0.295**	0.780	**0.428**	0.590
Set S4 (USF 40%)	Naïve Bayes	IDF	0.294	0.780	0.427	0.590
Set S4 (USF 40%)	LMT	N/A	**0.361**	0.847	**0.506**	**0.670**
Set S4 (USF 40%)	LMT	IDF	0.346	0.847	0.491	0.660
Set S4 (USF 40%)	SVM	N/A	0.331	**0.793**	0.468	0.620
Set S4 (USF 40%)	SVM	IDF	0.347	0.780	0.480	0.620

Table 15: Comparison of the positive class scores for models using Inverse Document Frequency and models with no feature selection

We highlighted in Table 15 the scores in which models with no feature selection outperformed the models using Inverse Document Frequency as feature selection. The imbalanced model using a SVM algorithm demonstrated a F-measure drop of ≈0.12 while the F-2 score drops ≈0.9. The LMT model at 40% also shows performance decrease, but with a drop of ≈0.1 in F-measure and F-2 score. This can also be seen in Figures 12 and 13, which summarize the scores for the models that applied undersampling combined with Inverse Document Frequency as a feature selection metric. A hypothesis to explain this overall reduction in performance caused by the Inverse Document Frequency filtering is that the frequent features, removed by this approach, might play an important role when combined with the infrequent features in the model design. For instance, some of the bioentities that were mapped as possibly discriminative features by the biocurators, are commonly found throughout the documents and therefore filtered out by this approach.

If we observe Tables 26 to 28, which used the sets of features S1, S2 and S3, we notice that many models present zero classification scores for the positive instances. In these specific cases, the positive class is completely overlooked by the classification models. The models output the predictions for all document instances as if they belong to the negative class. Generally, we identified two cases in which the models more affected by the Inverse Document Frequency: the first one is when the models were based on sets of features S1 and S2, and the second one is when the models were based

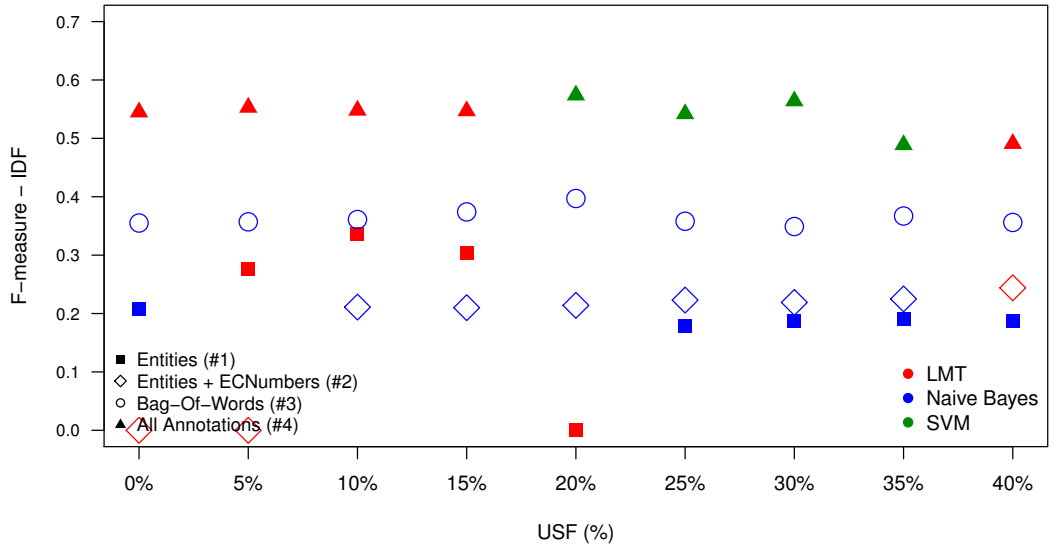

Figure 12: Summary of *mycoSORT* F-measure scores for the positive class. Best classifiers and set of features for each USF using Inverse Document Frequency

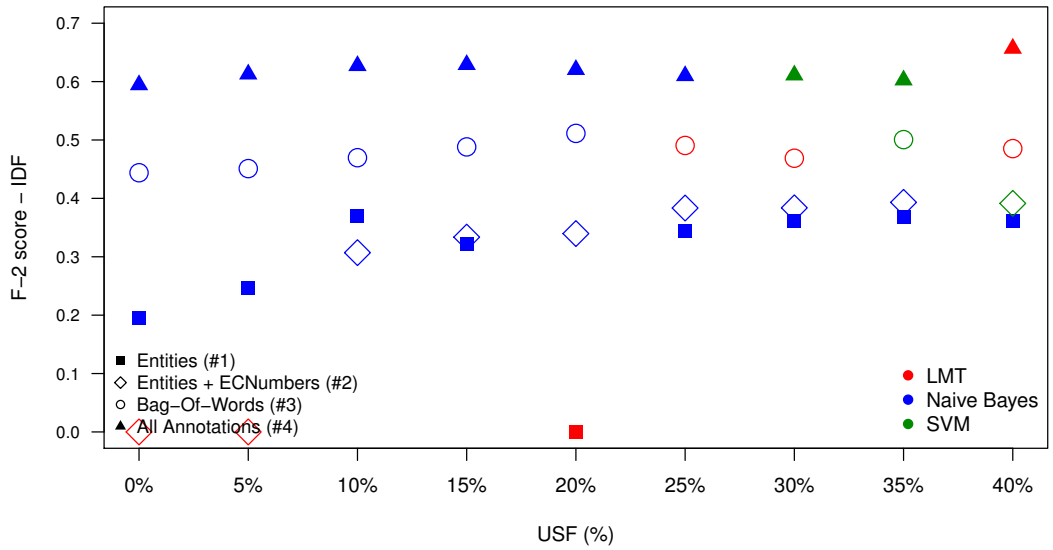

Figure 13: Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF using Inverse Document Frequency

on training sets generated using lower USFs (i.e., more imbalanced). We discuss now our hypotheses to explain this particular classification outcome in these two cases.

We first focus on the behavior of this filtering method related to these two sets of features, S1 and S2. The features composing sets S1 and S2 originally represent a small list, when compared to the other sets of features. In addition, S1 and S2 also have features that appear very often throughout the document instances, since they are respectively composed by 22 bioentities, and 22 bioentities plus EC numbers. After applying the Inverse Document Frequency and filter out the most common features, the subsets selected from sets S1 and S2 were even smaller, as shown in Table 14. Our assumption to explain this behavior is that the feature subsets selected from S1 and S2 was not composed by enough discriminative features to fit the classification models properly. Therefore, inefficient classification boundaries were built for the models based on S1 and S2, using the less discriminative feature subsets provided by the Inverse Document Frequency filtering, which leads to a specially worse performance in the positive instances.

Now, we focus on the performance of the Inverse Document Frequency with respect to the models with lower USFs. These models present a highly imbalanced class distribution. When computing the Inverse Document Frequency scores of features in an imbalanced training set, it is more likely to find uncommon features in negative documents, since they represent the majority of document instances. As the uncommon features receive higher scores when computing the Inverse Document Frequency, they were considered as relevant and kept for the model building. Our hypothesis is that the subset of features selected by this filtering method in imbalanced training sets had a stronger relation to the negative class. Hence, the use of these subsets introduced noise and more bias in the imbalanced models, making it harder for the classifier to output predictions for the positive class.

Another interesting observation we can make from analyzing Figure 12 is regarding the performance of models using the set of features S3 (circles). After applying Inverse Document Frequency, the S3 models seem to have dropped more in performance than the other models. Previously, S3 based models yield F-measures that were similar to the ones obtained by S4 (triangles), but since S3 based models use a naïve approach to extract features, apparently they are more impacted by the removal of the most common features. The decrease in the F-measure was higher than the lost of F-2 scores, indicating that the application of Inverse Document Frequency affected more the precision than the recall of models. This can be seen in Figure 14, where we evaluate the best performance obtained when using Inverse Document Frequency with regards to the baseline model. When comparing the best Inverse Document Frequency model with the TM1 model, we can observe that both approaches reach the same recall score, 0.847, while the precision in the Inverse Document Frequency model decreases from 0.361 to 0.346. The results obtained by the best model using Inverse Document Frequency, which we call Triage Model 2 (TM2), are highlighted in Table 29.

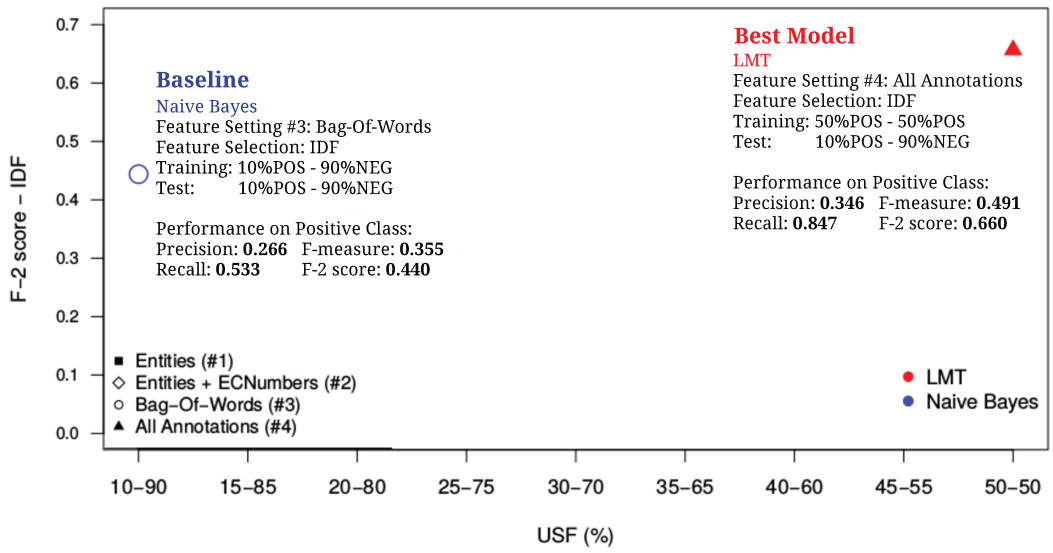

Figure 14: F-2 scores of baseline and best model for the approach using USFs and Inverse Document Frequency

The use of Odds Ratio (OR) filters out a greater number of features when compared to the number of features removed by the use of Inverse Document Frequency, as shown in Table 14. Again, the percentage of reduction is evaluated with respect to the number of features composing the models that did not apply feature selection methods. For the set S1, Odds Ratio reduces the number of features from 22 bioentities to 14-17 features. In set S2, the filter reduces the feature space size by ≈85% in the model with 0% USF to 83% in the model with 40% USF. For the set S3, Odds Ratio cuts down the number of features by ≈90% in the model with 0% USF to 88% in the model with 40% USF. Finally, for set S4 the reduction is ≈82.5% in the model with 0% USF to 80% in the model with 40% USF. If we look at the reduction in the feature space size of the sets S2, S3 and S4 made by Odds Ratio, we observe that the reduction percentage in the models with 0% USF to models with 40% USF varies in ≈2% to ≈3%. The reduction percentage of the feature space size observed in models with 0% USF to models with 40% USF yield by Inverted Document Frequency was of ≈47% for set S2, ≈87% for set S3 and ≈71% for set S4. This demonstrates that the reduction in the feature space size performed by Odds Ratio seems to be more consistent across the sets of features.

Even though Odds Ratio removes a much higher percentage of features, the results obtained by models using this metric yield better scores than the scores achieved by models using Inverse

Document Frequency filtering. To summarize the scores obtained by models applying Odds Ratio, we provide a comparison between the models with no feature selection and models using Odds Ratio in Table 16. These models are composed of the set of features S4 and the LMT classifier. To summarize our comparison, again we only provide the results using 0% USF and 40% USFs, and the complete results obtained by all USFs can be seen in Tables 21 and25. We highlighted the

Setting	**Classifier**	**Feature Selection**	**Precision**	**Recall**	**F-measure**	**F-2**
Set S4 (USF 0%)	Naïve Bayes	N/A	0.355	0.727	0.477	0.600
Set S4 (USF 0%)	Naïve Bayes	OR	0.326	**0.740**	0.453	0.590
Set S4 (USF 0%)	LMT	N/A	0.685	0.420	0.521	0.460
Set S4 (USF 0%)	LMT	OR	**0.706**	0.400	0.511	0.440
Set S4 (USF 0%)	SVM	N/A	0.867	0.087	0.158	0.110
Set S4 (USF 0%)	SVM	OR	0.826	**0.253**	**0.388**	**0.200**
Set S4 (USF 40%)	Naïve Bayes	N/A	0.295	0.780	0.428	0.590
Set S4 (USF 40%)	Naïve Bayes	OR	0.293	0.780	0.425	0.590
Set S4 (USF 40%)	LMT	N/A	0.361	0.847	0.506	0.670
Set S4 (USF 40%)	LMT	OR	**0.368**	**0.860**	**0.515**	**0.680**
Set S4 (USF 40%)	SVM	N/A	0.331	0.793	0.468	0.620
Set S4 (USF 40%)	SVM	OR	0.324	**0.833**	0.466	**0.630**

Table 16: Comparison of positive class scores of models using Odds Ratio and models with no feature selection

scores in which models using Odds Ratio outperformed models using no feature selection method in Table 16. In particular, for the models using 40% USF, recall increased ≈0.2 with the LMT and ≈0.4 with the SVM. F-measure and F-2 scores for the same LMT model raised ≈0.1, while the SVM model only improved F-2 score by ≈0.1.

In Figures 15 and 16 we demonstrate the F-measure and F-2 scores for the best models in each USF, with the use of Odds Ratio as a feature selection metric. As we can observe in Figure 15, the F-measure and F-2 scores curves of models using Odds Ratio are very similar to the curves generated by the models obtained without any feature selection, but with an overall improvement in performance. The Odds Ratio seems to provide better scores for the recall of S1 based models, which after the filtering are built with an even smaller set of 15 to 17 features. The best model identified within the models using Odds Ratio achieves better scores than the TM1 model. As we can observe in Figure 9, the best model using Odds Ratio reaches a similar score to TM1 model in precision (0.368), but a better score in recall, 0.860. The results obtained by the best model using Odds Ratio, which we call Triage Model 3 (TM3), are highlighted in Table 25.

As explained in Section 4.1, recall is a very important measure in our context since we are specially concerned with the ability of the model to identify the highest number of positive instances, as not to miss potential relevant information. We understand that the better performance of the models

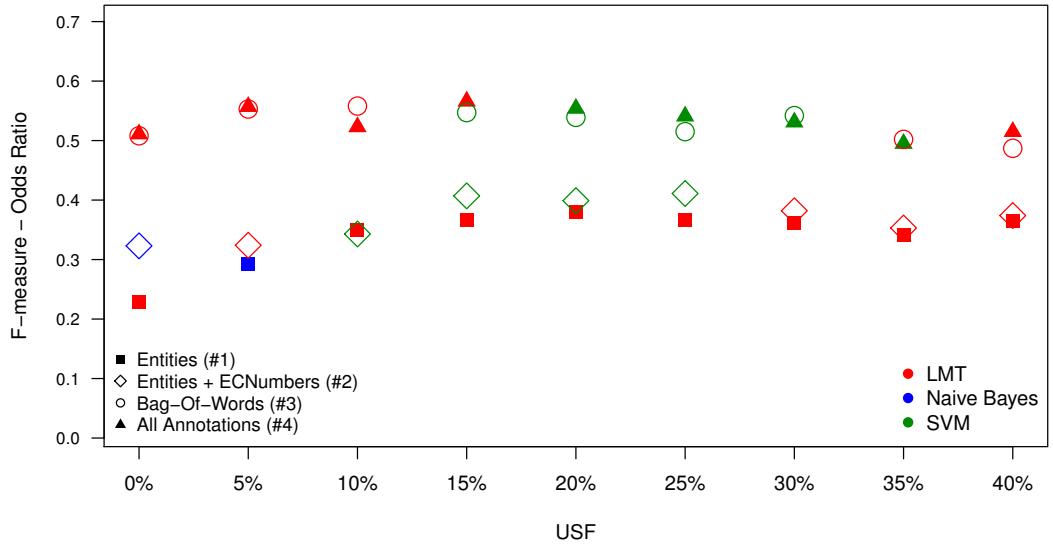

Figure 15: Summary of *mycoSORT* F-measure for the positive class. Best classifiers and set of features for each USF using Odds Ratio

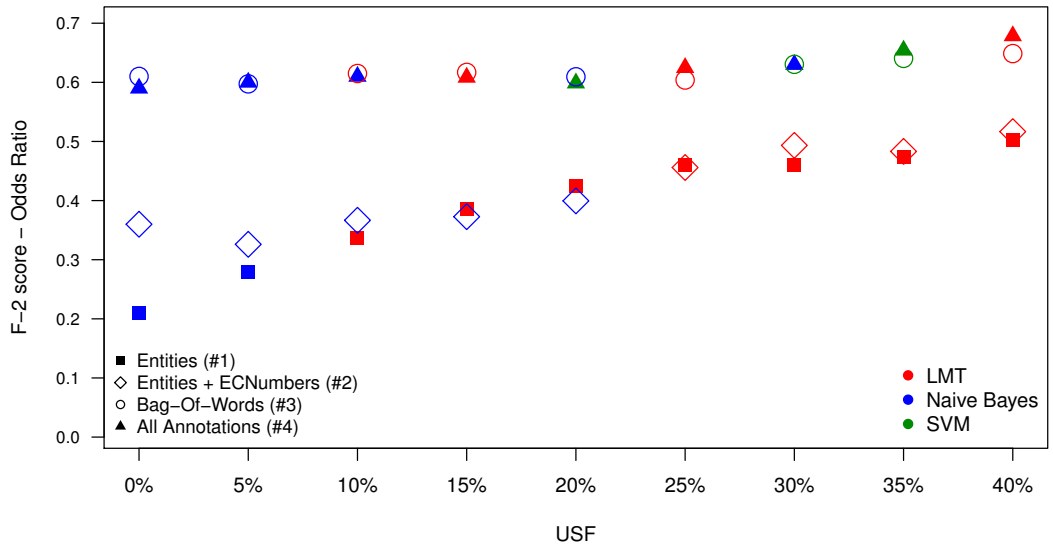

Figure 16: Summary of *mycoSORT* F-2 scores for the positive class. Best classifiers and set of features for each USF using Odds Ratio

using Odds Ratio is due to the fact that this measure uses as a filtering criteria the relationship between the occurrence of a feature and the occurrence of the positive class. Thus, the likelihood of selecting features with a stronger connection to the positive class is higher when using Odds Ratio than when using Inverse Document Frequency since the latter considers the entire document collection to compute the feature selection score for each attribute.

Model	**Precision**	**Recall**	**F-measure**	**F-2**
Baseline	0.307	0.720	0.430	0.570
TM1	**0.361**	**0.847**	**0.506**	**0.670**
TM2	0.346	0.847	0.491	0.660
TM3	**0.368**	**0.860**	**0.515**	**0.680**

Table 17: Comparison between best results obtained from the different model design approaches

Table 17 summarizes our findings after performing experiments with 324 different models. We list here the models TM1, TM2 and TM3, that achieved the best performances among the models based on the combination of the variables proposed in the Chapter 3 of this thesis. In order to better visualize the improvement in discriminative power gained with these three models, we compared their scores with the results we obtained with the baseline model. We identified that the two most fitting classification models to handle the triage task are TM1 and TM3.

In this chapter, we have presented the evaluation of our system. We first defined the metrics used to assess the performance of the various models tested. Then, we provided a detailed description of the experimental framework by defining the variables employed to build the models. Finally, we discussed the results, and analyzed our findings about the best sets of features, undersampling strategy and feature selection methods. In the next chapter, we will conclude by summarizing the most fitting approach for the proposed system to support the biomedical literature triage, and we will also discuss possible future work that could be carried out.

Chapter 5

Conclusion and Future Work

In this chapter we summarize our conclusions about the effectiveness of the methods studied in this thesis to handle the task of biomedical literature triage. First, we discuss the most important findings of our research, starting by the contribution of feature extraction techniques and the identification of the most relevant set of features for the task. Next, we re-iterate our findings about the imbalanced learning strategy developed for our experimental framework, and the use of undersampling as a method to tackle this issue. Finally, in Section 5.3, we present possibilities of future studies that can be conducted to improve this work, and continue the research of automatic support for biomedical literature triage.

5.1 Main Findings

In this thesis, we have developed an automatic approach to support the triage of biomedical documents. As we discussed in Chapter 2, the triage of scientific literature is a difficult task since it involves dealing with two important issues: the imbalanced class distribution in the data set, and the selection of relevant features to design the classification models. In order to propose a solution to support biomedical triage, we have studied a total of 324 classification models and have evaluated the influence of several variables on each of these models. These variables are composed by 4 sets of features, 2 feature selection metrics, 3 classification algorithms, and 9 undersampling factors (USFs).

5.1.1 Scientific Contributions

Best Models Overall Rather than aiming for high accuracy, we focused our attention in the model capability to correctly classifying the most interesting (positive) instances, which are the least represented in the document collection, and therefore the hardest to be predicted by the classifiers.

This issue, described in Chapter 2, could be compared to searching for a needle in a haystack. After the evaluation of 324 models, we have identified the two most fitting classification models to handle the triage task: TM1 and TM3. As described in Chapter 4, TM1 is composed of domain annotations (set of feature S4), combined with an equally balanced training set (40% USF) and the use of the LMT algorithm. TM3 has the same configuration as TM1, but with the use of Odds Ratio scores to filter the feature set. We have found that adopting the TM1 and TM3 models, as opposed to adopting a naïve or baseline model, results in an overall improvement in performance. In particular, compared to the baseline, TM1 raised precision scores from 0.307 to 0.361, recall scores from 0.720 to 0.847, F-measure from 0.430 to 0.506 and F-2 from 0.570 to 0.670. TM3 raised precision scores from 0.307 to 0.368, recall from 0.720 to 0.860, F-measure from 0.430 to 0.515, and F-2 from 0.570 to 0.680. This indicates that the techniques we studied and described in Chapter 2 are effective in terms of handling the main problems characteristic of the triage task.

We now focus on our findings concerning the best strategies to employ in order to design classification models that are capable of performing the difficult task of biomedical literature triage. First, we start with the interesting conclusion we draw from using different feature types to design the models. After analyzing the results obtained across the four sets of features utilized in our experiments, we found that although the naïve BOW approach for feature extraction demonstrates reasonable results, the set of features that provides the best results is the combination of domain-related annotations. This combination is composed by the 22 bioentities listed by biocurators as potential cues to identify relevant document instances, the annotated contents in the entity spans, the annotated contents in the sentence spans, as well as the EC numbers. Therefore, we suggest the incorporation of a group of domain-related attributes in the design of models intended to perform biomedical literature triage. In the context of our work, the domain-related annotations are provided by the mycoMINE text mining system. It is important to note that, even though mycoMINE was used to extract fungal-related bioentities from our document collection, the *mycoSORT* system is capable of handling different annotation schemes, in order to support the triage task in various biomedical research contexts. We address the reproducibility aspects of *mycoSORT* experiments with more details in Section 5.2.

Best Imbalanced Learning Strategy We now turn our attention to the techniques we explored to deal with the imbalanced class distribution, which directly affects the classifier performance, as described in Section 2.2. To define an effective imbalanced learning strategy, we evaluated the results of 3 different algorithms and the use of 9 USFs. This evaluation demonstrates how the use of undersampling not only significantly cuts down the quantity of data to be processed during the learning phase but also influences positively the model performance. As we observed in Section 4.4,

the classification scores generally increase with the progressive increment of USFs. Such behavior leads us to conclude that undersampling is an effective technique to be used in scenarios which the data is numerous and imbalanced.

Best Classification Algorithm The best results were provided by higher USFs combined with the use of the LMT algorithm. We understand that the better performance of LMT compared to the other algorithms is due to its properties to scale to more complex problems [Landwehr et al., 2005] since it relies on an elaborate combination of a tree structure that allows the algorithm to perform well when the data presents a nonlinear underlying distribution, and the logistic regression models that allows LMT to scale to larger and more complex datasets, where a regular decision tree would turn out larger and less accurate.

Best Feature Selection Method Finally, we discuss our findings regarding the application of feature selection methods to discard the least discriminative attributes from the sets of features. A comparison between all scores obtained by the use of feature selection pointed out that the best strategy in terms of feature space reduction and performance gain is to apply feature filtering according to Odds Ratio scores, with regards to the positive class. The detailed strategy employed to implement Odds Ratio selection is described in Section 3.4.2. By adopting Odds Ratio in the S4 based models, we created subsets of features that are up to 90% smaller than the original sets. At the same time, S4 based models using the Odds Ratio showed a general improvement in scores such as an increase of ≈19.4% in recall (from 0.72 to 0.86) and 19.8% in precision (from 0.307 to 0.368). In light of this observation, we assume that the use of feature selection is helpful to effectively narrow down the set of features used to create the models. Thus, we suggest the use of Odds Ratio as a metric to filter out the least discriminative features in the triage task.

5.1.2 Other Contributions

End-user Support The use of USFs in our imbalanced learning strategy, as well as the application of feature selection, resulted in a global reduction of the number of dataset instances used by the classifiers during the learning phase. This reduction implies that the task cost is cut down. By discarding a subset of the non relevant instances through undersampling, and discarding the least discriminative features through feature filtering, the matrix representation of the dataset is substantially smaller. For our context in the triage task, we are especially concerned about selecting an approach that provides satisfying performance, and also requires a somewhat low computational cost. The overall task cost, defined in terms of the time taken to fit the models, is important since eventually our goal is to provide *mycoSORT* end-users the choice of constantly improving the model

in case new relevant document instances are identified. This entails that the end-users must be able to incorporate new document instances to the training data, and re-train the model themselves, as well as output predictions for new test data, as frequently as the literature curation workflow requires.

Publications The research on biomedical literature triage conducted in this thesis gave rise to two research articles in the open access peer-reviewed scientific journals PLoS ONE [Almeida et al., 2014a] and DATABASE [Strasser et al., 2015]. Our work was also published in the Seventh International Biocuration Conference (ISB2014) [Almeida et al., 2014b], as well as in the workshop for Machine Learning for Clinical Data Analysis, Healthcare and Genomics at the Neural Information Processing Systems (NIPS 2014) conference [Almeida et al., 2014c].

The next section will discuss the availability and reproducibility of the *mycoSORT* system. We also explain the possibilities of applying our proposed solution to perform biomedical literature triage in different contexts, in which the document collection is related to subjects beyond the one addressed in this thesis.

5.2 Availability and Reproducibility

The experiments conducted in this thesis are completely reproducible. As a way to encourage further research in classification of biomedical text and support the literature triage of new research topics, we publicly released the *mycoSORT* system developed in our work and the document collection utilized in our evaluations. In this section we describe how *mycoSORT* and the fungal enzymes-related dataset can be accessed and used to reproduce our experiments and results, or how they can be applied as a tool to support other literature triage tasks.

The *mycoSORT* system is fully implemented and publicly released as an open source toolkit under the MIT License. The source code, along with general setup instructions, are accessible at the following address: `https://github.com/TsangLab/mycoSORT`. Together with *mycoSORT*, we also provided the corpus with all positive and negative document instances labeled during the mycoCLAP database manual curation. The corpus is released as a list of pairs containing: [document PubMed ID - document label].

The availability of *mycoSORT* allows the triage task to be reproduced in different research contexts. As we explained in Section 4.4, domain annotations were identified as the most discriminative features to compose the classification models designed for the biomedical literature triage. In our

context, we employed the open-source mycoMINE[1] text mining system to extract 22 specific bioentities from the document instances. To support the curation process of different research topics and allow *mycoSORT* to scale to application of domain features in the triage of new literature, we suggest the use of several wide-ranging annotation schemes [Aronson, 2001] [Ruch, 2006]. Some examples of schemes are the Medical Subject Headings (MeSH) vocabulary, the Gene Ontology (GO) and the Unified Medical Language System (UMLS) thesaurus. These schemes can be used to provide broad-spectrum biomedical annotations in scientific documents, which might comprise an extensive variety of research subjects.

5.3 Future Work

We describe in this section potential aspects to continue the research on biomedical literature triage, as well as opportunities we have identified to investigate possible improvements in the approaches and model designs proposed in this thesis.

Analysis of new datasets In our work, we have performed our experiments on a document collection related to the manual curation of the mycoCLAP database to generate the *mycoSet* dataset. This collection contains document instances related to fungal-enzymes, and it formed the basis of the development and evaluation of the classification models. An interesting analysis would be to utilize a different biomedical dataset, in order to evaluate the *mycoSORT* system performance, and if possible, obtain a human validation of the relevant document instances outputted by the system predictions. We are currently performing a similar evaluation of *mycoSORT* with the use of a new dataset. As an attempt to support the triage of scientific literature with bacteria-related content, we have submitted 6,658 unlabeled instances to be classified by one of the best models for the task, TM1. *mycoSORT* outputted a positive prediction for 980 instances, a number that represents ≈14.7% of the dataset. The ratio of relevant instances identified in the bacteria dataset seems encouraging, since it is similar to the proportion of relevant instances found in the fungal enzymes triage. However, as an ongoing work, biocurators are manually validating *mycoSORT* positive predictions for the bacteria dataset.

Study of other undersampling approaches The undersampling technique utilized in our work to handle the imbalanced data provides performance improvement, and can be easily incorporated in the *mycoSORT* pipeline. However, as the process of discarding document instances belonging to the majority class is executed randomly, the technique is rather naïve, and could be improved by adopting a more informed evaluation criterion before discarding an instance, as some approaches that

[1] https://github.com/TsangLab/Annotators/tree/master/mycoMINE

were described in Section 2.2.2. During our experiments, we attempted to perform undersampling using a stratified random approach. Indeed, the documents in *mycoSet* can be grouped by different enzyme families. The occurrence of different enzyme families in *mycoSet* was used as a reference for the stratified sample size and determine the number of documents that should be selected per enzyme family to compose a training corpus, such that the enzyme family would be represented in the training set in the same proportion as it appears in the entire document collection. The classification results on the training corpora generated by stratified undersampling demonstrated that the enzyme families are not as discriminative as we expected. Therefore, we decided to carry on our experimentations applying only a random selection approach. Independently of the informed technique chosen to perform the dataset undersampling, the computational resources must be taken into account, so as not to increase the overall task cost, as in the time taken to fit the classification model.

Evaluation of different classifiers As explained in Section 3.5, the classification algorithms used to design our models were off-the-shelf implementations provided by the Weka workbench. According to our findings, the classifier that yield better scores was the LMT. However, according to previous work conducted on imbalanced data learning, pointed out that the SVM algorithm should be capable of producing satisfying results. In our evaluations, SVM showed performance scores that were somewhat close, but overall lower, to the scores demonstrated by LMT. Therefore, a more detailed investigation of the performance of SVM, perhaps non-standard SVM implementations, such as the ones described in Section 2.4, would be promising for the literature triage task.

Exploration of different feature selection methods The feature selection approaches adopted in our work demonstrates that the use of a selection metric capable of accounting for the relation between the features and the occurrence of the relevant class provided a better outcome, when compared to approaches not using feature selection. According to the results obtained in our experiments, the application of Odds Ratio filtering results in the best discriminative feature subset. Thus, we recommend the investigation of other feature selection approaches to perform filtering, that will take into account the correlation between pairs of values, as the occurrence of a feature and the occurrence of the relevant class. Some possible metrics to be incorporated in the pipeline are correlation methods, such as the Pearson Correlation Coefficient (PCC), and dependence measures, such as Mutual Information (MI), which were compared in the work of [Guyon and Elisseeff, 2003].

Bibliography

R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Machine Learning: Proceedings of the European Conference on Machine Learning (ECML) 2004, September 20-24, Pisa, Italy*, pages 39–50. Springer, 2004.

H. Almeida, M.-J. Meurs, L. Kosseim, G. Butler, and A. Tsang. Machine Learning for Biomedical Literature Triage. *PLoS ONE*, 9(12):e115892, 2014a.

H. Almeida, M.-J. Meurs, L. Kosseim, G. Butler, and A. Tsang. A Machine Learning Approach for mycoCLAP Triage. In *Proceedings of the 7th International Biocuration Conference, April 6-9*, page 65, Toronto, ON, 2014b. International Society for Biocuration.

H. Almeida, M.-J. Meurs, L. Kosseim, G. Butler, and A. Tsang. Biomedical Literature Triage using Supervised Learning. In *NIPS 2014 - ML4CHG, 2nd Workshop on Machine Learning for Clinical Data Analysis, Healthcare and Genomics, December 8-12*, Montreal, QC, 2014c. Neural Information Processing Systems Foundation.

T. A. Almeida, J. Almeida, and A. Yamakami. Spam filtering: How the dimensionality reduction affects the accuracy of Naive Bayes classifiers. *Journal of Internet Services and Applications*, 1 (3):183–200, 2011.

K. H. Ambert and A. M. Cohen. K-information gain scaled nearest neighbors: a novel approach to classifying protein-protein interaction-related documents. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(1):305–310, 2012.

M.-L. Antonie, O. R. Zaiane, and A. Coman. Application of data mining techniques for medical image classification. In *Proceedings of the Second International Workshop on Multimedia Data Mining, (MDM/KDD) 2001, August 26, 2001, San Francisco, CA, USA*, pages 94–101, 2001.

C. N. Arighi, Z. Lu, M. Krallinger, K. B. Cohen, J. W. Wilbur, A. Valencia, L. Hirschman, and C. H. Wu. Overview of the BioCreative III Workshop. *BMC Bioinformatics*, 12(Suppl 8):S1, 2011.

C. N. Arighi, B. Carterette, K. B. Cohen, M. Krallinger, W. J. Wilbur, P. Fey, R. Dodson, L. Cooper, C. E. Van Slyke, W. Dahdul, et al. An overview of the BioCreative 2012 Workshop Track III: Interactive text mining task. *Database*, 2013:bas056, 2013.

C. N. Arighi, C. H. Wu, K. B. Cohen, L. Hirschman, M. Krallinger, A. Valencia, Z. Lu, J. W. Wilbur, and T. C. Wiegers. BioCreative-IV virtual issue. *Database*, 2014:bau039, 2014.

A. R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. In *Proceedings of the AMIA Symposium, November 3-7, 2001, Washington, DC, USA*, pages 17–21. American Medical Informatics Association, 2001.

P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics*, 16(5):412–424, 2000.

T. Basu and C. Murthy. Effective text classification by a supervised feature selection approach. In *Proceedings of the IEEE 12th International Conference on Data Mining Workshops (ICDMW), December 10, Brussels, Belgium*, pages 918–925. IEEE, 2012.

R. Batuwita and V. Palade. A new performance measure for class imbalance learning. Application to bioinformatics problems. In *International Conference on Machine Learning and Applications, ICMLA, December 13-15, Miami, Florida, USA*, pages 545–550. IEEE, 2009.

R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, pages 235–249, 2002.

L. Borrajo, R. Romero, E. L. Iglesias, and C. R. Marey. Improving imbalanced scientific text classification using sampling strategies and dictionaries. *Journal of Integrative Bioinformatics*, 8: 176, 2011.

C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Proceedings of Advances in Knowledge Discovery and Data Mining (KDD-2009), April 27-30, Bangkok, Thailand*, pages 475–482. Springer, 2009.

C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. DBSMOTE: Density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36(3):664–684, 2012.

D. Campos, J. Lourenço, S. Matos, and J. L. Oliveira. Egas: A collaborative and interactive document curation platform. *Database*, 2014:bau048, 2014.

E. Charton, M.-J. Meurs, L. Jean-Louis, and M. Gagnon. Using Collaborative Tagging for Text Classification: From Text Classification to Opinion Mining. *Informatics*, 1(1):32–51, 2013.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling TEchnique. *Journal of Artificial Intelligence Research*, 16:341–378, 2002.

C. Chen, A. Liaw, and L. Breiman. Using Random Forest to learn imbalanced data. Technical Report 666, Statistics Department, University of California, 2004.

G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18, 2006.

C. Drummond and R. C. Holte. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling. In *Workshop on Learning from Imbalanced Datasets II at the International Conference on Machine Learning (ICML-2003), Washington DC*, pages 1–8, 2003.

C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), August 4-10, Washington, DC, USA*, volume 17, pages 973–978, 2001.

A. Estabrooks, T. Jo, and N. Japkowicz. A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence*, 20(1):18–36, 2004. ISSN 1467-8640. doi: 10.1111/j.0824-7935.2004.t01-1-00228.x. URL `http://dx.doi.org/10.1111/j.0824-7935.2004.t01-1-00228.x`.

T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

A.-C. Haury, P. Gestraud, and J.-P. Vert. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, 6(12):e28210, 2011.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

H. He and Y. Ma. *Imbalanced Learning: Foundations, Algorithms and Applications*, chapter 1,2. IEEE Press, 2013.

L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. Overview of BioCreative: Critical assessment of information extraction for biology. *BMC bioinformatics*, 6(Suppl 1):S1, 2005.

L. Hirschman, G. A. C. Burns, M. Krallinger, C. hihi, K. B. Cohen, A. Valencia, C. H. Wu, A. Chatr-Aryamontri, K. G. Dowell, E. Huala, A. Lourenco, R. Nash, A.-L. Veuthey, T. Wiegers, and A. G. Winter. Text mining for the biocuration workflow. *Database*, 2012.

T. R. Hoens and N. V. Chawla. *Imbalanced Datasets: From Sampling to Classifiers*, pages 43–59. John Wiley & Sons, Inc., 2013.

D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre, S. Twigger, O. White, and S. Y. Rhee. Big data: The future of biocuration. *Nature*, 455(7209):47–50, 2008.

Y.-Y. Hsu and H.-Y. Kao. CoIN: A network analysis for document triage. *Database*, 2013:bat076, 2013.

L. Hunter and K. B. Cohen. Biomedical language processing: Perspective what is beyond PubMed? *Molecular cell*, 21(5):589, 2006.

N. Japkowicz. The Class Imbalance Problem: Significance and Strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI), Vancouver BC, Canada*, pages 111–117, 2000.

N. Japkowicz and M. Shah. *Evaluating learning algorithms: A classification perspective*. Cambridge University Press, 2011.

A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. EnsMart: A generic system for fast and flexible access to biological data. *Genome research*, 14(1):160–169, 2004.

S. Kim and W. J. Wilbur. Classifying protein-protein interaction articles using word and syntactic features. *BMC bioinformatics*, 12(Suppl 8):S9, 2011.

M. Krallinger, M. Vazquez, F. Leitner, D. Salgado, A. Chatr-aryamontri, A. Winter, L. Perfetto, L. Briganti, L. Licata, M. Iannuccelli, L. Castagnoli, G. Cesareni, M. Tyers, G. Schneider, F. Rinaldi, R. Leaman, G. Gonzalez, S. Matos, S. Kim, W. J. Wilbur, L. Rocha, H. Shatkay, A. V. Tendulkar, S. Agarwal, F. Liu, X. Wang, R. Rak, K. Noto, C. Elkan, Z. Lu, R. I. Dogan, J.-F. Fontaine, M. A. Andrade-Navarro, and A. Valencia. The Protein-Protein Interaction tasks of

BioCreative III: Classification/ranking of articles and linking bio-ontology concepts to full text. *BMC bioinformatics*, 12(Suppl 8):S3, 2011.

D. Kwon, S. Kim, S.-Y. Shin, A. Chatr-aryamontri, and W. J. Wilbur. Assisting manual literature curation for protein–protein interactions using BioQRator. *Database*, 2014:bau067, 2014.

N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005.

J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on Artificial Intelligence in Medicine, July 1-4, Cascais, Portugal*, volume 2101, pages 63–66. Springer, 2001.

H. Liu and H. Motoda. *Computational methods of feature selection*. CRC Press, Boca Raton FL, USA, 2007.

H. Liu, H. Motoda, R. Setiono, and Z. Zhao. Feature Selection: An Ever Evolving Frontier in Data Mining. In *Proceedings of the 4th Workshop on Feature Selection in Data Mining, June 21, Hyderabad, India*, pages 4–13, 2010.

Y. Liu, N. V. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech & Language*, 20(4): 468–494, 2006.

O. Loyola-González, M. García-Borroto, M. A. Medina-Pérez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and G. De Ita. An Empirical Study of Oversampling and Undersampling Methods for LCMine an Emerging Pattern Based Classifier. In *Pattern Recognition*, pages 264–273. Springer, 2013.

Z. Lu and L. Hirschman. Biocuration workflows and text mining: Overview of the BioCreative 2012 Workshop Track II. *Database*, 2012:bas043, 2012.

J. Luengo, A. Fernández, S. García, and F. Herrera. Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing*, 15(10):1909–1936, 2011.

J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, et al. Performance measures for information extraction. In *Proceedings of the DARPA Broadcast News Workshop, February 28-March 3, Herndon, Virginia, USA*, pages 249–252, 1999.

M. A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of the ICML-2003 Workshop on learning from imbalanced data sets II, Washington, DC*, volume 2, 2003.

S. Marsland. *Machine Learning: An algorithm perspective*. Chapman and Hall, 1 edition, 2009.

S. Matis-Mitchell, P. Roberts, C. O. Tudor, and C. Arighi IV. BioCreative IV interactive task. In *Proceedings of BioCreative IV, October 7-9, Bethesda, Maryland, USA*, volume 1, 2013.

B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

K. McCarthy, B. Zabar, and G. Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st International Workshop on Utility-based Data Mining, August 21, Chicago, Illinois, USA*, pages 69–77. ACM, 2005.

M.-J. Meurs, C. Murphy, I. Morgenstern, G. Butler, J. Powlowski, A. Tsang, and R. Witte. Semantic text mining support for lignocellulose research. *BMC Medical Informatics and Decision Making*, 12, 2012.

B. D. Morris and E. P. White. The EcoData Retriever: Improving Access to Existing Ecological Data. *PloS one*, 8(6):e65848, 2013.

A. Mountassir, H. Benbrahim, and I. Berrada. An empirical study to address the problem of unbalanced data sets in sentiment classification. *IEEE Systems, Man, Cybernetics*, pages 3298–3303, 2012.

U. S. Mudunuri, M. Khouja, S. Repetski, G. Venkataraman, A. Che, B. T. Luke, F. P. Girard, and R. M. Stephens. Knowledge and Theme Discovery across Very Large Biological Data Sets Using Distributed Queries: A Prototype Combining Unstructured and Structured Data. *PloS one*, 8 (12):e80503, 2013.

C. Murphy, J. Powlowski, M. Wu, G. Butler, and A. Tsang. Curation of characterized glycoside hydrolases of Fungal origin. *Database*, vol.2011:bar020, 2011.

National Center for Biotechnology Information. PubMed, 2005a.

National Center for Biotechnology Information. PubMed [Table, Stopwords], 2005b.

Y. Peng, Z. Wu, and J. Jiang. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43(1):15–23, 2010.

C. Quan, M. Wang, and F. Ren. An Unsupervised Text Mining Method for Relation Extraction from Biomedical Literature. *PloS one*, 9(7):e102039, 2014.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

M. M. Rahman and D. Davis. Addressing the Class Imbalance Problem in Medical Datasets. *International Journal of Machine Learning and Computing*, 3(2):224–228, 2013.

E. Ramentol, Y. Caballero, R. Bello, and F. Herrera. SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and information systems*, 33(2):245–265, 2012.

B. Raskutti and A. Kowalczyk. Extreme re-balancing for SVMs: A case study. *ACM SIGKDD Explorations Newsletter*, 6(1):60–69, 2004.

R. Romero, A. S. Vieira, E. Iglesias, and L. Borrajo. BioClass: A Tool for Biomedical Text Classification. In *Proceedings of the 8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014), June 4-6, Salamanca, Spain*, pages 243–251. Springer, 2014.

P. Ruch. Automatic assignment of biomedical categories: Toward a generic approach. *Bioinformatics*, 22(6):658–664, 2006.

Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

R. N. Smith, J. Aleksic, D. Butano, A. Carr, S. Contrino, F. Hu, M. Lyne, R. Lyne, A. Kalderimis, K. Rutherford, R. Stepan, J. Sullivan, M. Wakeling, X. Watkins, and G. Micklem. InterMine: A flexible data warehouse system for the integration and analysis of heterogeneous biological data. *Bioinformatics*, 28(23):3163–3165, 2012.

K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

K. Strasser, E. McDonnell, C. Nyaga, M. Wu, S. Wu, H. Almeida, M.-J. Meurs, L. Kosseim, J. Powlowski, G. Butler, and A. Tsang. mycoCLAP, the database for characterized lignocellulose-active proteins of fungal origin: resource and text mining curation support. *Database*, 2015, 2015.

C.-T. Su and Y.-H. Hsiao. An evaluation of the robustness of MTS for imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 19(10):1321–1332, 2007.

M. Szumilas. Explaining odds ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 19(3):227, 2010.

Y. Tang and Y.-Q. Zhang. Granular SVM with repetitive undersampling for highly imbalanced protein homology prediction. In *Proceedings of the IEEE International Conference on Granular Computing, 2006*, pages 457–460. IEEE, 2006.

Y. Tang, B. Jin, and Y.-Q. Zhang. Granular support vector machines with association rules mining for protein homology prediction. *Artificial Intelligence in Medicine*, 35(1):121–134, 2005.

Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. SVMs modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: IEEE Transactions on Cybernetics*, 39(1): 281–288, 2009.

V. N. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.

C. G. Varassin, A. Plastino, H. C. Leitáo, and B. Zadrozny. Undersampling strategy based on clustering to improve the performance of splice site classification in human genes. In *Proceedings of the 24th International Workshop on Database and Expert Systems Applications, August 26-29, Prague, Czech Republic*, pages 85–89. IEEE Computer Society, 2013.

B. C. Wallace, T. A. Trikalinos, J. Lau, C. Brodley, and C. H. Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC bioinformatics*, 11(1):55, 2010.

M. Wang, W. Zhang, W. Ding, D. Dai, H. Zhang, H. Xie, L. Chen, Y. Guo, and J. Xie. Parallel Clustering Algorithm for Large-Scale Biological Data Sets. *PloS one*, 9(4):e91315, 2014.

M. Wasikowski and X.-W. Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1388–1400, 2010.

E. C. Webb. Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes, 1992.

G. M. Weiss. Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

G. M. Weiss. Foundations of Imbalanced Learning. *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 13–41, 2013.

G. M. Weiss and F. Provost. The effect of class distribution on classifier learning: an empirical study. *Department of Computer Science, Rutgers University*, 2001.

G. M. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proceedings of the 2007 International Conference on Data Mining, June 25-28, Las Vegas, Nevada, USA*, pages 35–41, 2007.

Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning, July 8-12, Nashville, Tennessee, USA*, volume 97, pages 412–420, 1997.

D.-J. Yu, J. Hu, Z.-M. Tang, H.-B. Shen, J. Yang, and J.-Y. Yang. Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. *Neurocomputing*, 104:180–190, 2013.

H. Zhang, M. Huang, and X. Zhu. Protein-protein interaction extraction from bio-literature with compact features and data sampling strategy. In *Proceedings of the 4th International Conference on Biomedical Engineering and Informatics (BMEI), October 15-17, Shanghai, China*, volume 4, pages 1767–1771. IEEE, 2011.

G. K. Zipf. Selected studies of the principle of relative frequency in language. Harvard University Press, Cambridge, Massachusetts, 1932.

Appendix A

Dataset Queries

Queries submitted to PubMed that were used to compose the *mycoSet* corpus.

```
acetylxylan esterase AND fung*
alpha-1,2-mannosidase AND fung*
alpha-amylase AND fung*
alpha-galactosidase AND fung*
alpha-glucosidase AND fung*
alpha-L-rhamnosidase AND fung*
aryl-alcohol oxidase AND fung*
avenacinase AND fung*
beta-glucosidase AND fung*
beta-mannanase AND fung*
Beta-xylosidase AND fung*
cellobiohydrolase AND fung*
chitin deacetylase AND fung*
Chitosanase AND fung*
cutinase AND fung*
dextranase AND fung*
Endo-1,6-beta-glucanase AND fung*
endo-beta-1,3-galactanase AND fung*
xyloglucanase AND fung*
endo-polygalacturonase AND fung*
endoglucanase AND fung*
glucose oxidase AND fung*
glucoamylase AND fung*
glyoxal oxidase AND fung*
Hexosaminidase AND fung*
isopullulanase AND fung*
laminarinase AND fung*
licheninase AND fung*
lignin peroxidase AND fung*
manganese peroxidase AND fung*
mixed-link glucanase AND fung*
mutanase AND fung*
oligo-1,6-glucosidase AND fung*
pectate lyase AND fung*
pectin lyase AND fung*
pectin methylesterase AND fung*
rhamnogalacturonan hydrolase AND fung*
pyranose 2-oxidase AND fung*
peroxidase AND fung*
rhamnogalacturonan acetylesterase AND fung*
rhamnogalacturonan lyase AND fung*
tomatinase AND fung*
```

exo-1,3-beta-glucanase AND fung*	trehalase AND fung*
exo-arabinanase AND fung*	versatile peroxidase AND fung*
exo-polygalacturonase AND fung*	xylanase AND fung*
feruloyl esterase AND fung*	xylogalacturonase AND fung*
galactanase AND fung*	Endo-N-acetyl-beta-D-glucosaminidase AND fung*

Appendix B

mycoSORT Experimental Results

Table 18: *mycoSORT* results using USFs for set S1

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0%)	Naïve Bayes	0.286	0.227	0.253	0.182	0.240
Training set (USF 0%)	LMT	0.492	0.207	0.291	0.274	0.230
Training set (USF 0%)	SVM	0.714	0.033	0.064	0.140	0.04
Training set (USF 5%)	Naïve Bayes	0.294	0.280	0.287	0.210	0.280
Training set (USF 5%)	LMT	0.461	0.233	0.310	0.278	0.260
Training set (USF 5%)	SVM	0.645	0.133	0.221	0.264	0.160
Training set (USF 10%)	Naïve Bayes	0.269	0.307	0.287	0.202	0.300
Training set (USF 10%)	LMT	0.376	0.213	0.272	0.226	0.230
Training set (USF 10%)	SVM	0.47	0.207	0.287	0.264	0.230
Training set (USF 15%)	Naïve Bayes	0.301	0.347	0.322	0.241	0.340
Training set (USF 15%)	LMT	0.352	0.413	0.380	0.307	0.400
Training set (USF 15%)	SVM	0.387	0.287	0.330	0.271	0.300
Training set (USF 20%)	Naïve Bayes	0.263	0.340	0.297	0.209	0.320
Training set (USF 20%)	LMT	0.348	0.480	0.403	0.331	0.450
Training set (USF 20%)	SVM	0.353	0.353	0.353	0.281	0.350
Training set (USF 25%)	Naïve Bayes	0.243	0.353	0.288	0.197	0.320
Training set (USF 25%)	LMT	0.286	0.547	0.375	0.301	0.460
Training set (USF 25%)	SVM	0.282	0.413	0.335	0.251	0.380
Training set (USF 30%)	Naïve Bayes	0.277	0.440	0.340	0.257	0.390
Training set (USF 30%)	LMT	0.291	0.627	0.397	0.334	0.510
Training set (USF 30%)	SVM	0.258	0.48	0.336	0.252	0.410
Training set (USF 35%)	Naïve Bayes	0.242	0.440	0.312	0.223	0.380
Training set (USF 35%)	LMT	0.233	0.620	0.338	0.266	0.470
Training set (USF 35%)	SVM	0.210	0.633	0.316	0.241	0.450
Training set (USF 40%)	Naïve Bayes	0.254	0.467	0.329	0.243	0.400
Training set (USF 40%)	LMT	0.269	0.660	0.382	0.321	0.510
Training set (USF 40%)	SVM	0.196	0.667	0.303	0.229	0.450

Results of only the Positive Class using set of features S1

Table 19: *mycoSORT* results using USFs for set S2

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0%)	Naïve Bayes	0.285	0.380	0.326	0.242	0.360
Training set (USF 0%)	LMT	0.516	0.107	0.177	0.202	0.130
Training set (USF 0%)	SVM	1.000	0.020	0.039	0.134	0.020
Training set (USF 5%)	Naïve Bayes	0.273	0.373	0.315	0.230	0.350
Training set (USF 5%)	LMT	0.426	0.173	0.246	0.224	0.200
Training set (USF 5%)	SVM	0.833	0.033	0.064	0.155	0.040
Training set (USF 10%)	Naïve Bayes	0.268	0.427	0.329	0.243	0.380
Training set (USF 10%)	LMT	0.412	0.233	0.298	0.255	0.260
Training set (USF 10%)	SVM	0.688	0.073	0.133	0.203	0.090
Training set (USF 15%)	Naïve Bayes	0.268	0.427	0.329	0.243	0.380
Training set (USF 15%)	LMT	0.398	0.300	0.342	0.284	0.320
Training set (USF 15%)	SVM	0.604	0.193	0.293	0.306	0.220
Training set (USF 20%)	Naïve Bayes	0.275	0.440	0.338	0.255	0.390
Training set (USF 20%)	LMT	0.322	0.393	0.354	0.276	0.380
Training set (USF 20%)	SVM	0.471	0.327	0.386	0.338	0.350
Training set (USF 25%)	Naïve Bayes	0.258	0.507	0.342	0.260	0.420
Training set (USF 25%)	LMT	0.321	0.520	0.397	0.324	0.460
Training set (USF 25%)	SVM	0.364	0.420	0.390	0.318	0.410
Training set (USF 30%)	Naïve Bayes	0.237	0.540	0.329	0.248	0.430
Training set (USF 30%)	LMT	0.328	0.500	0.396	0.322	0.450
Training set (USF 30%)	SVM	0.323	0.473	0.384	0.308	0.430
Training set (USF 35%)	Naïve Bayes	0.227	0.513	0.315	0.229	0.410
Training set (USF 35%)	LMT	0.267	0.587	0.367	0.295	0.470
Training set (USF 35%)	SVM	0.251	0.573	0.349	0.274	0.460
Training set (USF 40%)	Naïve Bayes	0.244	0.520	0.332	0.250	0.420
Training set (USF 40%)	LMT	0.267	0.707	0.388	0.334	0.530
Training set (USF 40%)	SVM	0.217	0.613	0.321	0.245	0.450

Results of only the Positive Class using set of features S2

Table 20: *mycoSORT* results using USFs for set S3

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0%)	**Naïve Bayes**	**0.307**	**0.720**	**0.430**	**0.382**	**0.570**
Training set (USF 0%)	LMT	0.656	0.420	0.512	0.485	0.450
Training set (USF 0%)	SVM	0.833	0.033	0.064	0.155	0.040
Training set (USF 5%)	Naïve Bayes	0.310	0.733	0.436	0.390	0.580
Training set (USF 5%)	LMT	0.600	0.500	0.545	0.503	0.520
Training set (USF 5%)	SVM	0.703	0.173	0.278	0.319	0.200
Training set (USF 10%)	Naïve Bayes	0.307	0.760	0.438	0.396	0.590
Training set (USF 10%)	LMT	0.574	0.567	0.570	0.523	0.570
Training set (USF 10%)	SVM	0.704	0.333	0.452	0.449	0.370
Training set (USF 15%)	Naïve Bayes	0.309	0.793	0.445	0.41	0.600
Training set (USF 15%)	LMT	0.458	0.693	0.552	0.504	0.630
Training set (USF 15%)	SVM	0.596	0.413	0.488	0.451	0.440
Training set (USF 20%)	Naïve Bayes	0.314	0.793	0.450	0.415	0.610
Training set (USF 20%)	LMT	0.422	0.653	0.513	0.460	0.590
Training set (USF 20%)	SVM	0.545	0.527	0.536	0.485	0.530
Training set (USF 25%)	Naïve Bayes	0.312	0.780	0.446	0.408	0.600
Training set (USF 25%)	LMT	0.399	0.673	0.501	0.449	0.590
Training set (USF 25%)	SVM	0.481	0.580	0.526	0.470	0.560
Training set (USF 30%)	Naïve Bayes	0.288	0.767	0.418	0.377	0.580
Training set (USF 30%)	LMT	0.388	0.727	0.506	0.461	0.620
Training set (USF 30%)	SVM	0.460	0.687	0.551	0.503	0.630
Training set (USF 35%)	Naïve Bayes	0.302	0.780	0.435	0.397	0.590
Training set (USF 35%)	LMT	0.359	0.807	0.497	0.465	0.650
Training set (USF 35%)	SVM	0.369	0.800	0.505	0.472	0.650
Training set (USF 40%)	Naïve Bayes	0.303	0.773	0.435	0.396	0.590
Training set (USF 40%)	LMT	0.344	0.840	0.488	0.463	0.650
Training set (USF 40%)	SVM	0.338	0.840	0.482	0.456	0.650

Results of only the Positive Class using set of features S3

Table 21: *mycoSORT* results using USFs for set S4

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0%)	Naïve Bayes	0.355	0.727	0.477	0.431	0.600
Training set (USF 0%)	LMT	0.685	0.420	0.521	0.498	0.460
Training set (USF 0%)	SVM	0.867	0.087	0.158	0.257	0.110
Training set (USF 5%)	Naïve Bayes	0.365	0.740	0.489	0.446	0.610
Training set (USF 5%)	LMT	0.585	0.480	0.527	0.484	0.500
Training set (USF 5%)	SVM	0.729	0.287	0.411	0.424	0.330
Training set (USF 10%)	Naïve Bayes	0.349	0.787	0.484	0.448	0.630
Training set (USF 10%)	LMT	0.552	0.600	0.575	0.526	0.590
Training set (USF 10%)	SVM	0.670	0.420	0.516	0.491	0.450
Training set (USF 15%)	Naïve Bayes	0.342	0.787	0.477	0.441	0.620
Training set (USF 15%)	LMT	0.478	0.647	0.550	0.498	0.600
Training set (USF 15%)	SVM	0.607	0.473	0.532	0.491	0.490
Training set (USF 20%)	Naïve Bayes	0.342	0.793	0.478	0.443	0.630
Training set (USF 20%)	LMT	0.425	0.64	0.511	0.456	0.580
Training set (USF 20%)	SVM	0.521	0.587	0.552	0.500	0.570
Training set (USF 25%)	Naïve Bayes	0.322	0.787	0.457	0.421	0.610
Training set (USF 25%)	LMT	0.389	0.747	0.511	0.469	0.630
Training set (USF 25%)	SVM	0.474	0.667	0.554	0.504	0.620
Training set (USF 30%)	Naïve Bayes	0.336	0.773	0.469	0.430	0.610
Training set (USF 30%)	LMT	0.398	0.780	0.527	0.490	0.650
Training set (USF 30%)	SVM	0.459	0.673	0.546	0.496	0.620
Training set (USF 35%)	Naïve Bayes	0.304	0.800	0.440	0.406	0.600
Training set (USF 35%)	LMT	0.343	0.760	0.473	0.433	0.610
Training set (USF 35%)	SVM	0.357	0.793	0.493	0.458	0.640
Training set (USF 40%)	Naïve Bayes	0.295	0.780	0.428	0.389	0.590
Training set (USF 40%)	**LMT**	**0.361**	**0.847**	**0.506**	**0.481**	**0.670**
Training set (USF 40%)	SVM	0.331	0.793	0.468	0.433	0.620

Results of only the Positive Class using set of features S4

Table 22: *mycoSORT* results using USFs for set S1 + Odds Ratio filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + OR)	Naïve Bayes	0.250	0.207	0.226	0.150	0.210
Training set (USF 0% + OR)	LMT	0.524	0.147	0.229	0.240	0.170
Training set (USF 0% + OR)	SVM	1.000	0.007	0.013	0.077	0.010
Training set (USF 5% + OR)	Naïve Bayes	0.323	0.267	0.292	0.223	0.280
Training set (USF 5% + OR)	LMT	0.450	0.180	0.257	0.238	0.200
Training set (USF 5% + OR)	SVM	0.690	0.133	0.223	0.276	0.160
Training set (USF 10% + OR)	Naïve Bayes	0.255	0.360	0.298	0.209	0.330
Training set (USF 10% + OR)	LMT	0.377	0.327	0.350	0.284	0.340
Training set (USF 10% + OR)	SVM	0.431	0.187	0.260	0.235	0.210
Training set (USF 15% + OR)	Naïve Bayes	0.249	0.380	0.301	0.211	0.340
Training set (USF 15% + OR)	LMT	0.337	0.400	0.366	0.290	0.390
Training set (USF 15% + OR)	SVM	0.301	0.247	0.271	0.200	0.260
Training set (USF 20% + OR)	Naïve Bayes	0.290	0.327	0.307	0.226	0.320
Training set (USF 20% + OR)	LMT	0.324	0.460	0.380	0.304	0.420
Training set (USF 20% + OR)	SVM	0.329	0.327	0.328	0.253	0.330
Training set (USF 25% + OR)	Naïve Bayes	0.233	0.413	0.298	0.206	0.360
Training set (USF 25% + OR)	LMT	0.274	0.553	0.366	0.292	0.460
Training set (USF 25% + OR)	SVM	0.276	0.367	0.315	0.230	0.340
Training set (USF 30% + OR)	Naïve Bayes	0.224	0.480	0.305	0.215	0.390
Training set (USF 30% + OR)	LMT	0.268	0.560	0.362	0.287	0.460
Training set (USF 30% + OR)	SVM	0.249	0.473	0.326	0.241	0.400
Training set (USF 35% + OR)	Naïve Bayes	0.225	0.473	0.305	0.216	0.390
Training set (USF 35% + OR)	LMT	0.232	0.640	0.341	0.272	0.470
Training set (USF 35% + OR)	SVM	0.182	0.573	0.276	0.185	0.400
Training set (USF 40% + OR)	Naïve Bayes	0.281	0.433	0.341	0.258	0.390
Training set (USF 40% + OR)	LMT	0.249	0.673	0.364	0.303	0.500
Training set (USF 40% + OR)	SVM	0.191	0.680	0.299	0.226	0.450

Results of only the Positive Class using set of features S1 and Odds Ratio as feature selection

Table 23: *mycoSORT* results using USFs for set S2 + Odds Ratio filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + OR)	Naïve Bayes	0.253	0.487	0.333	0.248	0.410
Training set (USF 0% + OR)	LMT	0.256	0.693	0.374	0.317	0.520
Training set (USF 0% + OR)	SVM	0.207	0.673	0.317	0.248	0.460
Training set (USF 5% + OR)	Naïve Bayes	0.217	0.580	0.316	0.235	0.430
Training set (USF 5% + OR)	LMT	0.244	0.640	0.353	0.286	0.480
Training set (USF 5% + OR)	SVM	0.207	0.573	0.304	0.22	0.420
Training set (USF 10% + OR)	Naïve Bayes	0.226	0.573	0.324	0.245	0.440
Training set (USF 10% + OR)	LMT	0.277	0.613	0.382	0.315	0.490
Training set (USF 10% + OR)	SVM	0.285	0.487	0.360	0.280	0.430
Training set (USF 15% + OR)	Naïve Bayes	0.259	0.400	0.314	0.226	0.360
Training set (USF 15% + OR)	LMT	0.325	0.507	0.396	0.322	0.460
Training set (USF 15% + OR)	SVM	0.356	0.487	0.411	0.340	0.450
Training set (USF 20% + OR)	Naïve Bayes	0.292	0.440	0.351	0.270	0.400
Training set (USF 20% + OR)	LMT	0.360	0.387	0.373	0.301	0.380
Training set (USF 20% + OR)	SVM	0.411	0.387	0.399	0.334	0.390
Training set (USF 25% + OR)	Naïve Bayes	0.293	0.400	0.338	0.256	0.370
Training set (USF 25% + OR)	LMT	0.365	0.307	0.333	0.268	0.320
Training set (USF 25% + OR)	SVM	0.558	0.320	0.407	0.377	0.350
Training set (USF 30% + OR)	Naïve Bayes	0.275	0.400	0.326	0.241	0.370
Training set (USF 30% + OR)	LMT	0.391	0.227	0.287	0.241	0.250
Training set (USF 30% + OR)	SVM	0.648	0.233	0.343	0.353	0.270
Training set (USF 35% + OR)	Naïve Bayes	0.280	0.340	0.307	0.223	0.330
Training set (USF 35% + OR)	LMT	0.500	0.240	0.324	0.299	0.270
Training set (USF 35% + OR)	SVM	0.739	0.113	0.197	0.266	0.140
Training set (USF 40% + OR)	Naïve Bayes	0.278	0.387	0.323	0.238	0.360
Training set (USF 40% + OR)	LMT	0.567	0.113	0.189	0.222	0.130
Training set (USF 40% + OR)	SVM	0.500	0.007	0.013	0.049	0.010

Results of only the Positive Class using set of features S2 and Odds Ratio as feature selection

Table 24: *mycoSORT* results using USFs for set S3 + Odds Ratio filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + OR)	Naïve Bayes	0.321	0.780	0.454	0.417	0.610
Training set (USF 0% + OR)	LMT	0.698	0.400	0.508	0.491	0.440
Training set (USF 0% + OR)	SVM	0.814	0.233	0.363	0.409	0.270
Training set (USF 5% + OR)	Naïve Bayes	0.313	0.773	0.445	0.406	0.600
Training set (USF 5% + OR)	LMT	0.608	0.507	0.553	0.511	0.520
Training set (USF 5% + OR)	SVM	0.641	0.393	0.488	0.461	0.430
Training set (USF 10% + OR)	Naïve Bayes	0.316	0.787	0.450	0.414	0.610
Training set (USF 10% + OR)	LMT	0.483	0.660	0.558	0.508	0.610
Training set (USF 10% + OR)	SVM	0.622	0.460	0.529	0.492	0.490
Training set (USF 15% + OR)	Naïve Bayes	0.313	0.807	0.451	0.418	0.610
Training set (USF 15% + OR)	LMT	0.438	0.687	0.535	0.486	0.620
Training set (USF 15% + OR)	SVM	0.547	0.547	0.547	0.496	0.550
Training set (USF 20% + OR)	Naïve Bayes	0.312	0.800	0.449	0.415	0.610
Training set (USF 20% + OR)	LMT	0.432	0.653	0.520	0.467	0.590
Training set (USF 20% + OR)	SVM	0.494	0.593	0.539	0.486	0.570
Training set (USF 25% + OR)	Naïve Bayes	0.300	0.767	0.432	0.391	0.580
Training set (USF 25% + OR)	LMT	0.407	0.687	0.511	0.461	0.600
Training set (USF 25% + OR)	SVM	0.437	0.627	0.515	0.460	0.580
Training set (USF 30% + OR)	Naïve Bayes	0.321	0.787	0.456	0.419	0.610
Training set (USF 30% + OR)	LMT	0.402	0.713	0.514	0.468	0.620
Training set (USF 30% + OR)	SVM	0.440	0.707	0.542	0.496	0.630
Training set (USF 35% + OR)	Naïve Bayes	0.287	0.807	0.424	0.390	0.590
Training set (USF 35% + OR)	LMT	0.373	0.767	0.502	0.463	0.630
Training set (USF 35% + OR)	SVM	0.362	0.793	0.497	0.462	0.640
Training set (USF 40% + OR)	Naïve Bayes	0.294	0.82	0.433	0.402	0.600
Training set (USF 40% + OR)	LMT	0.344	0.833	0.487	0.460	0.650
Training set (USF 40% + OR)	SVM	0.330	0.827	0.471	0.443	0.640

Results of only the Positive Class using set of features S3 and Odds Ratio as feature selection

Table 25: *mycoSORT* results using USFs for set S4 + Odds Ratio filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + OR)	Naïve Bayes	0.326	0.740	0.453	0.409	0.590
Training set (USF 0% + OR)	LMT	0.706	0.400	0.511	0.495	0.440
Training set (USF 0% + OR)	SVM	0.826	0.253	0.388	0.430	0.290
Training set (USF 5% + OR)	Naïve Bayes	0.334	0.753	0.463	0.421	0.600
Training set (USF 5% + OR)	LMT	0.561	0.553	0.557	0.508	0.550
Training set (USF 5% + OR)	SVM	0.604	0.367	0.456	0.427	0.400
Training set (USF 10% + OR)	Naïve Bayes	0.321	0.780	0.454	0.417	0.610
Training set (USF 10% + OR)	LMT	0.475	0.580	0.523	0.466	0.560
Training set (USF 10% + OR)	SVM	0.570	0.460	0.509	0.464	0.480
Training set (USF 15% + OR)	Naïve Bayes	0.320	0.773	0.453	0.414	0.600
Training set (USF 15% + OR)	LMT	0.508	0.640	0.566	0.516	0.610
Training set (USF 15% + OR)	SVM	0.535	0.553	0.544	0.493	0.550
Training set (USF 20% + OR)	Naïve Bayes	0.318	0.753	0.448	0.405	0.590
Training set (USF 20% + OR)	LMT	0.418	0.627	0.501	0.445	0.570
Training set (USF 20% + OR)	SVM	0.492	0.633	0.554	0.502	0.600
Training set (USF 25% + OR)	Naïve Bayes	0.324	0.760	0.454	0.413	0.600
Training set (USF 25% + OR)	LMT	0.385	0.740	0.507	0.464	0.620
Training set (USF 25% + OR)	SVM	0.455	0.667	0.541	0.490	0.610
Training set (USF 30% + OR)	Naïve Bayes	0.344	0.793	0.480	0.445	0.630
Training set (USF 30% + OR)	LMT	0.422	0.707	0.529	0.482	0.620
Training set (USF 30% + OR)	SVM	0.430	0.693	0.531	0.482	0.620
Training set (USF 35% + OR)	Naïve Bayes	0.296	0.767	0.428	0.387	0.580
Training set (USF 35% + OR)	LMT	0.357	0.767	0.487	0.448	0.620
Training set (USF 35% + OR)	SVM	0.352	0.833	0.495	0.468	0.650
Training set (USF 40% + OR)	Naïve Bayes	0.293	0.780	0.425	0.387	0.590
Training set (USF 40% + OR)	**LMT**	**0.368**	**0.860**	**0.515**	**0.493**	**0.680**
Training set (USF 40% + OR)	SVM	0.324	0.833	0.466	0.439	0.630

Results of only the Positive Class using set of features S4 and Odds Ratio as feature selection

Table 26: *mycoSORT* results using USFs for set S1 + Inverse Document Frequency filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + IDF)	Naïve Bayes	0.235	0.187	0.208	0.132	0.190
Training set (USF 0% + IDF)	LMT	0.500	0.007	0.013	0.049	0.010
Training set (USF 0% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 5% + IDF)	Naïve Bayes	0.324	0.233	0.271	0.208	0.250
Training set (USF 5% + IDF)	LMT	0.419	0.207	0.277	0.242	0.230
Training set (USF 5% + IDF)	SVM	0.682	0.100	0.174	0.237	0.120
Training set (USF 10% + IDF)	Naïve Bayes	0.250	0.420	0.313	0.225	0.370
Training set (USF 10% + IDF)	LMT	0.381	0.300	0.336	0.274	0.310
Training set (USF 10% + IDF)	SVM	0.429	0.160	0.233	0.216	0.180
Training set (USF 15% + IDF)	Naïve Bayes	0.227	0.360	0.278	0.184	0.320
Training set (USF 15% + IDF)	LMT	0.306	0.300	0.303	0.226	0.300
Training set (USF 15% + IDF)	SVM	0.481	0.167	0.248	0.241	0.190
Training set (USF 20% + IDF)	Naïve Bayes	0.000	0.000	0.000	0.000	0.000
Training set (USF 20% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 20% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 25% + IDF)	Naïve Bayes	0.100	0.887	0.179	-0.003	0.340
Training set (USF 25% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 25% + IDF)	SVM	0.500	0.007	0.013	0.049	0.010
Training set (USF 30% + IDF)	Naïve Bayes	0.104	0.947	0.188	0.045	0.360
Training set (USF 30% + IDF)	LMT	0.000	0.000	0.000	-0.012	0.000
Training set (USF 30% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 35% + IDF)	Naïve Bayes	0.106	0.967	0.190	0.061	0.370
Training set (USF 35% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 35% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 40% + IDF)	Naïve Bayes	0.104	0.947	0.188	0.045	0.360
Training set (USF 40% + IDF)	LMT	0.104	0.933	0.187	0.038	0.360
Training set (USF 40% + IDF)	SVM	0.104	0.947	0.188	0.045	0.360

Results of only the Positive Class using set of features S1 and Inverse Document Frequency as feature selection

Table 27: *mycoSORT* results using USFs for set S2 + Inverse Document Frequency filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + IDF)	Naïve Bayes	0.000	0.000	0.000	0.000	0.000
Training set (USF 0% + IDF)	LMT	0.000	0.000	0.000	-0.017	0.000
Training set (USF 0% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 5% + IDF)	Naïve Bayes	0.000	0.000	0.000	0.000	0.000
Training set (USF 5% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 5% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 10% + IDF)	Naïve Bayes	0.139	0.440	0.211	0.088	0.310
Training set (USF 10% + IDF)	LMT	0.385	0.033	0.061	0.089	0.040
Training set (USF 10% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 15% + IDF)	Naïve Bayes	0.129	0.553	0.210	0.085	0.330
Training set (USF 15% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 15% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 20% + IDF)	Naïve Bayes	0.132	0.560	0.214	0.092	0.340
Training set (USF 20% + IDF)	LMT	0.180	0.060	0.090	0.050	0.070
Training set (USF 20% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 25% + IDF)	Naïve Bayes	0.131	0.740	0.223	0.118	0.380
Training set (USF 25% + IDF)	LMT	0.221	0.153	0.181	0.110	0.160
Training set (USF 25% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 30% + IDF)	Naïve Bayes	0.128	0.767	0.219	0.113	0.380
Training set (USF 30% + IDF)	LMT	0.141	0.427	0.212	0.089	0.300
Training set (USF 30% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 35% + IDF)	Naïve Bayes	0.131	0.787	0.225	0.128	0.390
Training set (USF 35% + IDF)	LMT	0.134	0.593	0.218	0.100	0.350
Training set (USF 35% + IDF)	SVM	0.132	0.060	0.083	0.024	0.070
Training set (USF 40% + IDF)	Naïve Bayes	0.113	0.853	0.199	0.075	0.370
Training set (USF 40% + IDF)	LMT	0.151	0.633	0.244	0.145	0.390
Training set (USF 40% + IDF)	SVM	0.130	0.787	0.223	0.124	0.390

Results of only the Positive Class using set of features S2 and Inverse Document Frequency as feature selection

Table 28: *mycoSORT* results using USFs for set S3 + Inverse Document Frequency filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + IDF)	Naïve Bayes	0.266	0.533	0.355	0.277	0.440
Training set (USF 0% + IDF)	LMT	0.000	0.000	0.000	0.000	0.000
Training set (USF 0% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 5% + IDF)	Naïve Bayes	0.265	0.547	0.357	0.280	0.450
Training set (USF 5% + IDF)	LMT	0.333	0.007	0.013	0.035	0.010
Training set (USF 5% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 10% + IDF)	Naïve Bayes	0.261	0.587	0.361	0.289	0.470
Training set (USF 10% + IDF)	LMT	0.143	0.007	0.013	0.010	0.010
Training set (USF 10% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 15% + IDF)	Naïve Bayes	0.269	0.613	0.374	0.306	0.490
Training set (USF 15% + IDF)	LMT	0.283	0.200	0.234	0.168	0.210
Training set (USF 15% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 20% + IDF)	Naïve Bayes	0.289	0.633	0.397	0.333	0.510
Training set (USF 20% + IDF)	LMT	0.277	0.480	0.351	0.270	0.420
Training set (USF 20% + IDF)	SVM	0.000	0.000	0.000	0.000	0.000
Training set (USF 25% + IDF)	Naïve Bayes	0.254	0.607	0.358	0.288	0.470
Training set (USF 25% + IDF)	LMT	0.246	0.653	0.357	0.292	0.490
Training set (USF 25% + IDF)	SVM	1.000	0.013	0.026	0.110	0.020
Training set (USF 30% + IDF)	Naïve Bayes	0.249	0.580	0.349	0.274	0.460
Training set (USF 30% + IDF)	LMT	0.209	0.680	0.320	0.253	0.470
Training set (USF 30% + IDF)	SVM	0.406	0.087	0.143	0.151	0.100
Training set (USF 35% + IDF)	Naïve Bayes	0.260	0.627	0.367	0.300	0.490
Training set (USF 35% + IDF)	LMT	0.178	0.673	0.282	0.204	0.430
Training set (USF 35% + IDF)	SVM	0.185	0.873	0.305	0.268	0.500
Training set (USF 40% + IDF)	Naïve Bayes	0.249	0.620	0.356	0.286	0.480
Training set (USF 40% + IDF)	LMT	0.187	0.807	0.304	0.253	0.490
Training set (USF 40% + IDF)	SVM	0.165	0.88	0.278	0.232	0.470

Results of only the Positive Class using set of features S3 and Inverse Document Frequency as feature selection

Table 29: *mycoSORT* results using USFs for set S4 + Inverse Document Frequency filtering

Undersampling(USF)	***Classifier***	***Precision***	***Recall***	***F-measure***	***MCC***	***F-2***
Training set (USF 0% + IDF)	Naïve Bayes	0.350	0.720	0.471	0.424	0.590
Training set (USF 0% + IDF)	LMT	0.717	0.440	0.545	0.526	0.480
Training set (USF 0% + IDF)	SVM	0.750	0.020	0.039	0.112	0.020
Training set (USF 5% + IDF)	Naïve Bayes	0.363	0.740	0.487	0.443	0.610
Training set (USF 5% + IDF)	LMT	0.591	0.520	0.553	0.508	0.530
Training set (USF 5% + IDF)	SVM	0.733	0.147	0.244	0.302	0.170
Training set (USF 10% + IDF)	Naïve Bayes	0.346	0.787	0.481	0.445	0.630
Training set (USF 10% + IDF)	LMT	0.524	0.573	0.548	0.496	0.560
Training set (USF 10% + IDF)	SVM	0.659	0.387	0.487	0.465	0.420
Training set (USF 15% + IDF)	Naïve Bayes	0.339	0.800	0.476	0.443	0.630
Training set (USF 15% + IDF)	LMT	0.478	0.640	0.547	0.495	0.600
Training set (USF 15% + IDF)	SVM	0.576	0.453	0.507	0.464	0.470
Training set (USF 20% + IDF)	Naïve Bayes	0.336	0.787	0.471	0.435	0.620
Training set (USF 20% + IDF)	LMT	0.427	0.660	0.518	0.466	0.600
Training set (USF 20% + IDF)	SVM	0.545	0.607	0.574	0.525	0.590
Training set (USF 25% + IDF)	Naïve Bayes	0.321	0.787	0.456	0.419	0.610
Training set (USF 25% + IDF)	LMT	0.359	0.720	0.479	0.432	0.600
Training set (USF 25% + IDF)	SVM	0.489	0.607	0.542	0.488	0.580
Training set (USF 30% + IDF)	Naïve Bayes	0.332	0.773	0.465	0.427	0.610
Training set (USF 30% + IDF)	LMT	0.404	0.700	0.512	0.464	0.610
Training set (USF 30% + IDF)	SVM	0.500	0.647	0.564	0.514	0.610
Training set (USF 35% + IDF)	Naïve Bayes	0.299	0.807	0.436	0.403	0.600
Training set (USF 35% + IDF)	LMT	0.334	0.727	0.458	0.412	0.590
Training set (USF 35% + IDF)	SVM	0.372	0.713	0.489	0.441	0.600
Training set (USF 40% + IDF)	Naïve Bayes	0.294	0.78	0.427	0.389	0.590
Training set (USF 40% + IDF)	**LMT**	**0.346**	**0.847**	**0.491**	**0.467**	**0.660**
Training set (USF 40% + IDF)	SVM	0.347	0.780	0.480	0.444	0.620

Results of only the Positive Class using set of features S4 and Inverse Document Frequency as feature selection